Final Report

DESIGN OF A FAULT TOLERANT AIRBORNE DIGITAL COMPUTER

Volume II — Computational Requirements and Technology

By: R. S. RATNER, E. B. SHAPIRO, H. M. ZEIDLER, S. E. WAHLSTROM, C. B. CLARK, and J. GOLDBERG

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LANGLEY RESEARCH CENTER
HAMPTON, VIRGINIA 23365

NICHOLAS MURRAY — CONTRACT MONITOR

CONTRACT NAS1-10920

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

Final Report *October 1973*

DESIGN OF A FAULT TOLERANT AIRBORNE DIGITAL COMPUTER

Volume II — Computational Requirements and Technology

By: R. S. RATNER, E. B. SHAPIRO, H. M. ZEIDLER, S. E. WAHLSTROM, C. B. CLARK, and J. GOLDBERG

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LANGLEY RESEARCH CENTER
HAMPTON, VIRGINIA 23365

NICHOLAS MURRAY — CONTRACT MONITOR

CONTRACT NAS1-10920

SRI Project 1406

Approved by:

D. R. BROWN, *Director*
Information Science Laboratory

BONNAR COX, *Executive Director*
Information Science and Engineering Division

ABSTRACT

This final report summarizes the work at Stanford Research Institute on the design of a fault tolerant digital computer for aircraft. Volume II of the report is composed of two parts; Part 1 is concerned with the computational requirements associated with an advanced commercial aircraft. Part 2 reviews the technology that will be available for the implementation of the computer in the 1975-1985 period. With regard to the computation task we have categorized 26 computations according to computational load, memory requirements, criticality, permitted down-time, and the need to save data in order to effect a roll-back. The technology part stresses the impact of large scale integration (LSI) on the realization of logic and memory. We also consider module interconnection possibilities so as to minimize fault propagation. Volume I of this report presents preliminary designs of three candidate architecture, that are suitable for the aircraft computational environment.

PART 1

COMPUTATIONAL REQUIREMENTS

TABLE OF CONTENTS FOR PART 1

ILLUSTRATIONS

TABLES

PREFACE

This report, issued in two volumes, summarizes the work of Stanford Research Institute on Contract NAS1-10920. The goal of the contract was to specify the design of a computer, destined for use as the central computer in an advanced, high-performance commerical aircraft. Because of the critical nature of many of the computations, fault tolerance was the primary design goal of the computer. Other important design goals of the computer relate to

- The matching of the architecture to the aircraft computations
- The capability for expansion or contraction to meet the requirements of various missions
- The suitability to post 1975 technology.

Volume I is concerned with the architecture of fault tolerant computers, that are matched to the aircraft environment. We selected and studied three candidate architectures as part of Task I of the contract. Two of these architectures, Software Implemented Fault Tolerance (SIFT) and Bus Checker System (BUCS) are new and as such are described in detail. The third candidate architecture is a multiprocessor concept that is due to Al Hopkins of MIT-Draper Laboratories. We are aware of the extensive work that has been devoted to fault tolerant techniques and architectures over the past decade. However, a survey of this work has pointed out significant deficiencies in each architecture, for our particular constraints. For the most well-known of these previously studied architectures we document the deficiencies.

Volume II of the report is organized as two parts. Part 1 is concerned with the computational requirements of an aircraft, wherein it is assumed that all of the computations scattered among special purpose analogue and mechanical computers would be carried out by a centralized digital computer. In addition to the usual computations associated with a commercial aircraft, e.g. navigation, stability augmentation, decrab, we also assume advanced cockpit displays and fly-by-wire. These various computations are categorized according to criticality, and for each computation we derive such parameters as memory requirements, processor requirements, iteration rates, the tolerable down time and the amount of data that must be saved in the event of failure. These results are concisely tabulated.

Part 2 of Volume II is concerned with the technology for realizing the contral computer. It is assumed that production would commence in the late 1970's. The two aspects of the realization that we consider are concerned with logic and memory and with module interconnections. With regard to logic and memory we assess the various technologies, MOS, CMOS, BIPOLAR, etc., as a function of requirements of speed, reliability, cost, number of units. In addition, we discuss such realizations as customized large scale integrated(LSI), medium scale integrated(MSI), programmable logic in the light of the above requirements. With regard to interconnection technology the primary goal is to prevent the propagation of faults beyond some predetermined module boundaries. Various approaches toward achieving this fault confinement are assessed in terms of speed, cost,reliability.

We would like to acknowledge the support of Nick Murray and his colleagues at NASA-Langley--Sal Bavuso, Larry Spencer, Bill Dove, and Brian Lupton. They interacted with us on all phases of the project and provided valuable guidance. On the computation aspects many aircraft and avionic specialists provided us with detailed descriptions of algorithms as well as experience on the conversion of analogue algorithms to a digital representation. In particular, we would like to thank the people at Boeing, Collins Radio and NASA-Lewis Research Center who spent so much time with us. With regard to architecture, we have had stimulating discussions with Al Hopkins, Al Avizienis, Bill Carter, Bill Martin, Barry Borgerson and Jim Miller. Many of their ideas are reflected in our candidate architectures.

I INTRODUCTION AND BACKGROUND

This part of the report documents SRI's work on estimating computational requirements for an ultrareliable airborne digital computer for use in an Advanced Technology Transport in the 1975-1985 time period. This work constituted Task II of SRI's project on the design of a fault-tolerant airborne digital computer, being conducted under Contract NAS1-10920 with NASA Langley Research Center. The purpose of this contract is to develop an ultrareliable computer design, through use of fault tolerance, and the purpose of the task reported on here is to develop estimates of the computational requirements imposed by all contemporary and projected future avionics and aircraft functions. These estimates are to be used for determining the computational size and power required on the digital computer, as well as to specify other constraints such as reconfiguration time.

The scope of the computational requirements task is sharply delineated by the task's objective to provide necessary informational input to the computer design task. This scope includes researching various avionics and aircraft functions to the depth necessary to develop representative computational estimates for sizing the computer design. Development of refined, or "optimal," algorithms for digital computer implementation of such functions is beyond the scope of this effort. Assumptions and simplifications were made as necessary within the course of this study to stay within this scope. These assumptions represent qualifications on the generality of our results. The assumptions and simplifications are documented in this report to permit rederivation of computational estimates under different assumptions should there be in the future a change in the trend of evolution of aircraft and avionics functions from that perceived at the time of this writing.

Many of the aircraft functions included in our research are not now a part of any commercial aircraft. Some of these functions are in conceptual stages of development. Others have flown experimentally, often using an implementation other than a digital computer. Thus our estimates are derived in an environment that is clearly one of incomplete information. We reduced functional concepts to computational estimates by making whatever assumptions seemed necessary; we translated functions currently implemented by analog mechanical or logical means into digital computer computational estimates using engineering judgments and much information obtained in oral discussion with practicing avionics and aircraft system designers. Those individuals whom we contacted for information and their judgments are listed in Appendix A.

The remainder of this report is organized into three sections. The first of these, Section II, describes the way we have classified aircraft functions, and the definitions and methodology we have used to state reliability requirements for the airborne computation of these functions. Section III summarizes in tabular form the computational estimates that have resulted from our work. Finally, Section IV discusses functions individually to elaborate on the specific assumptions, sources of information, and character of computation.

II FUNCTION CLASSES, STUDY METHODOLOGY, AND RELIABILITY CONSIDERATIONS

For purposes of this study we have divided aircraft and avionics functions into five classes, as follows: (1) Attitude Control, (2) Flight Path Control, (3) Navigation, (4) Communications (including ATC interaction and collision avoidance consideration) & (5) Aircraft Systems Support Functions. Function Class 1, attitude control, includes in our definition conventional stability augmentation (considering the aircraft as a rigid body), and any desired handling qualities modification. Function Class 2 consists of the control loops that drive the attitude control system to achieve and maintain a desired flight path in three dimensions and in time. The navigation functions of Class 3 are those that enable determination of the desired flight path, using inertial, air data, and radio sensing for position and velocity determination. Class 4 includes communications between various distinct functions within the aircraft, and communications external to the aircraft, including cooperative collision avoidance, participation with air traffic control, and transfer of other noncontrol messages between air and ground. Class 5, a miscellaneous category, contains such important functions as that of operating the aircraft power systems, the integrated data sensing and recording function (AIDS) coming into use in commercial aircraft, and other system monitoring functions. The computational results table in Section III lists all those functions that we have considered in this study.

To permit the computer design to exercise fault tolerance in the most flexible manner to support the aircraft mission requires having some notion of priority among aircraft functions. The concept of criticality defined below serves this purpose. Within the context of the extent to which a function is critical to performance of the aircraft mission, we will define reliability requirements for these functions.

There are several component aspects of criticality. These are discussed below and a shorthand notation for referring to them is defined in subsequent tables:

(1) In iterative computation of any aircraft function, the possibility exists that through either a computer failure or a deliberate allocation of computing priorities, occasionally a prescribed computational iteration may not be performed. Depending on the aircraft function itself, this may have no noticeable effect on aircraft performance or may seriously degrade some aspect of performance. To permit the most flexibility in designing the computer, we define, on a function-by-function basis, the number of successive iterations that can be missed, on an occasional basis, without serious effect on performance of that aircraft system. This parameter is denoted as MISS, generally a small integer.

(2) Again, depending on the particular aircraft function, the degree of necessity of preserving correct working data (including current state information) in the event of a computational failure is an important computer design criterion. DATA, the "yes or no" parameter, specifies whether such data need to be protected or not.

(3) In the event that an aircraft function being implemented in a digital computer fails and cannot be brought to operating status (computation re-established) within MISS iterations, there are several possible situations that may exist, with various implications for the seriousness of the failure. We use the term <u>backup</u> to denote a substitute function, one that can serve in lieu of the failed computer-implemented function. For example a fly-by-wire attitude and flight path control system may have a mechanical linkage as backup, or one computer function may substitute for another; e.g., navigation may be continued using air data and radio information in the event of an inertial system failure. There are four possible situations:

(a) No backup inside or outside the computer.

(b) Backup only via another computer function.

(c) Backup only outside the computer.

(d) Backup both internally and externally.

BACK, the parameter that specifies which of these possibilities exist, has value a, b, c, or d, corresponding to the listing above.

(4) For our purposes we define reliability as the probability that a function will not be absent (not computed) for more than MISS iterations at one time during any one-hour period. It is felt that reliability required can be adequately described in terms of function classes rather than for each individual function. For this purpose we consider each function to be in a certain "functional criticality class." One reliability value (designated REL) will be estimated for each such class, for each possible condition of backup.

We recognize five levels of criticality, as follows: Criticality Level 1--A function immediately critical to the safety of flight, e.g., stability augmentation for an inherently unstable aircraft. Criticality Level 2--A function that will be critical to the safety of flight at some future time during the mission, e.g., altimetry or airspeed display. Criticality Level 3--A function whose loss requires a significant change in mission to avoid degradation of safety, e.g., gust alleviation stability augmentation failure requiring significant change of speed and altitude. Criticality Level 4--A function whose loss imposes substantial operational penalties on air crew or ATC, e.g., navigation or communication failure. Criticality Level 5--A function whose loss has undesirable economic consequences but no significant safety degradation or operational penalty, e.g., loss of airborne integrated data system function, engine trim control or active structural fatigue control. Obviously, reliability requirements will decrease with increasing criticality level. Table 2 in Section III shows on a function-by-function basis the values that we adopt for the parameters MISS, DATA, BACK, and criticality level. The necessary iteration rate for computation of each of these functions is indicated on this chart as well. The concept of allowing a few missed iterations without declaring a failure is unconventional at this time and deserves some comment on how we develop values for the MISS parameter. Simply stated, the estimated length of time that would have to elapse before a noticeable or dangerous condition developed, following failure of computation of any particular function, is stated in terms of the number of iterations it represents. For example, occasional loss of up to three iterations of the attitude control computation is considered tolerable. This represents 0.15 second at the iteration rate quoted, an interval too short for any untoward condition to develop.

Table 1 states the reliability requirements we adopt in terms of function criticality levels. The values listed in the table are failure probabilities, defined as one minus the probability that the function will not be absent for more than MISS successive iterations per hour of operation. We assume no repair in flight outside of the built-in fault tolerance for each function. A 4-hour mission is assumed, and all failures are assumed to be detected immediately. The following paragraphs described the rationale and assumptions that we used to generate reliability requirements.

TABLE 1

RELIABILITY REQUIREMENTS*

CRIT	BACK			
	a	b	c†	d
1	10^{-8}	2.5×10^{-4}	$>10^{-8}$	$>2.5 \times 10^{-4}$
2	10^{-8}	2.5×10^{-4}	$>10^{-8}$	$>2.5 \times 10^{-4}$
3	0.8×10^{-4}	2.5×10^{-4}	$>0.8 \times 10^{-4}$	$>2.5 \times 10^{-4}$
4	2.5×10^{-4}	2.5×10^{-4}	$>2.5 \times 10^{-4}$	$>2.5 \times 10^{-4}$
5	3.3×10^{-4}	3.3×10^{-4}	$>3.3 \times 10^{-4}$	$>3.3 \times 10^{-4}$

* Entries are (1-reliability).

† Entries under c depend on alternative computer load.

The basis for the reliability requirement that we have assumed, for purposes of this study, is the statistical fatal accident rate for the best year (1971) scheduled air carrier civil aviation statistics. This risk is about 10^{-6}/per-hours exposure, and is about the same as the risk of contracting a fatal disease. Making a conservative assumption that there will be no survivors from an accident due to loss of a criticality level 1 or 2 function, we have an equivalent event--a fatal aircraft accident--with the same probability, 10^{-6}/aircraft hours flown.

Sufficient information is not available to us to derive, through reliability budget considerations, a reliability requirement for the airborne digital computer. Therefore, we will use our own judgment as a crude estimate, realizing that this is little more than a guess at present. First, we will assume that 10% of the total fatal aircraft accidents are associated with failures within the airplane and power plant systems, and we will assume further that the avionics functions associated with the airborne digital computer are allowed to contribute 10% of this reliability budget component. Thus we have a reliability requirement of 10^{-8} per hour of operation for a level 1 or level 2 function with no internal or external backup (BACK Condition "a"). This represents a probability two orders of magnitude smaller than the best civil aviation statistics for overall fatal accident rates.

We recognize and point out here that the basis for our reliability requirement specification, described above, represents only the most cursory and first-cut sort of analysis. However, we have been unable to uncover up to this point any firmer basis. To our knowledge FAA has never publicly expressed an acceptable quantitative level of safety for the system. Thus we must be content for the present with the values assumed here and must recognize in our computer architecture design task the fact that this number may change in the future should more information become available.

For Criticality Levels 3, 4, and 5, respectively, we postulate an acceptable incidence of failure to be once per 3000 missions (a 4-hour mission is assumed), once per 1000 missions, and once per 3000 hours (750 missions) respectively. These numbers are rationalized as follows: For level 3, this means an unpleasant, less safe situation in an aircraft once in about ten years of operation, and for the frequent air traveler, once in several hundred years. For level 4, it means a severe operational penalty for a flight crew about once every three or four years. For level 5, assuming that a level 5 failure could eventually cost as much as $50,000, the number postulated would mean an increase in total costs, over the lifetime of the aircraft, in the neighborhood of 1 percent.

For the case of BACK condition "c," we assume that a function's failure would have the impact of a level 4 failure with BACK condition "a." Thus, levels 1, 2 and 3 requirements can be degraded to the level 4, BACK condition "a" requirement.

For BACK condition "b," for want of knowledge at this time concerning the impact on the computer's load of having to invoke a substitute, backup, function, we use (conservatively) the no-backup requirement of BACK condition "a." The impact of a failure of a function with backup condition "d" is evidently no worse than that of either a condition "b" or "c" function failure, so without further information on the impact of such a failure on crew workload or computer load, we use (again conservatively) the lessor reliability of those for conditions "b" or "c."

III COMPUTATIONAL RESULTS SUMMARY

This section presents in tabular form the individual and aggregate computational requirements that we have estimated for the avionics and aircraft functions considered. Table 2 is a tabulation of various computation-related parameters on a function-by-function basis. Table 3 presents, again on a function-by-function basis, the computational estimates in terms of memory and computer time required. Aggregate estimates for the assumed heavy load case of instrument approach and landing--under conditions of zero visibility (Category III-C)--appear at the end of this table. Notes related to the entries in Table 2 and 3 appear following these tables.

Certain qualifications and explanations must be presented along with these tables. These are the subject of the following paragraphs. First, the reader must bear in mind the scope prescribed for this computational requirements estimation task, namely, to provide estimates for sizing a computer complex, rather than to design refined avionics functions for digital implementation. Thus, while the numerical entries in these tables may be stated in terms of tenths of a percent, these values are merely the values arrived at from analysis of the particular examples studied. The conclusion one should draw is that the computational requirement of such a function is roughly equivalent to the value stated.

The numerical tabulations of Table 3 also deserve some explanation here. These estimates are derived in terms of a "reference" or "benchmark" computer with the following characteristics: Two types of operations are assumed, a long (multiply, divide) requiring 16 microseconds, and a short (load, store, add, subtract, test), requiring 4 microseconds. Instructions are 24 bits, including provisions for address. Data registers of both 16- and 32-bit sizes are assumed available. All such data are held in core or other high-speed memory. The reference machine contains a clock or other suitable provision for referencing real time.

The concept of tolerating the possibility of MISS computational iterations was adopted to provide maximum flexibility for computer reconfiguration in the event of failure. The numbers for this parameter, presented in Table 2, are merely informed judgments, not precise estimates. The only consideration used in arriving at these numbers was that the time interval represented should be short enough that no unpleasant aircraft behavior can occur, yet long enough to permit the computer to reconfigure itself so as to tolerate an internal fault. Similarly, our placement of aircraft and avionics functions in particular backup classes has been done on a judgmental basis, considering current practice, perceived trends in aircraft technology, and the advanced technology transport typical mission, a 4-hour, Mach 1, air-carrier-type flight, in a civil aviation context. Assignment of a criticality class number to each function was done on the same basis, using judgments made on the basis of our own aviation experience and that of airline and aircraft manufacturing people with whom we discussed the matter.

Entries in the column labeled "Data Protection" indicate the necessity of preserving working data for a particular function in the event of a computational failure. For example, recovery from a failure of a navigation computation usually requires having available the last computed position or velocity components. Therefore, these data must be protected from loss due to computational failure. On the other hand, recovery from active flutter control computation failure can be made solely with new sensor data as it is acquired. No working data need be protected.

The final column of parameter values in Table 2 lists the iteration rate assumed for each full computation of the particular function. These rates were adopted on the basis of a reasonable balance between current conservative commercial aviation design practice, trends in new aviation technology, and results of experimental avionics programs now in progress within the Government and aviation manufacturing industry. Some of the iteration rate entries define a range of iteration rates,

TABLE 2

RELIABILITY PARAMETERS

					Iteration
	MISSed	Data	BACKup	Criticality	Rate
	Iterations	Protection	Class	Class	(per second)
Function Classes 1 and 2 (Attitude Flight and Path Control)					
Digital attitude control (with stability augmentation)for flight conditions other than automatic landing	2-3	Yes	a	1	5,20
Active flutter control	2-3	No	a	1	250
Active gust maneuver load control	2-3	No	a	3 and/or 5	240
Automatic flight path control					
Automatic landing including ILS/Inertial/DME and attitude control during automatic landing	2-3	Yes	a or b	1	20 (Horiz. Calc.) 160 (Vert. Calc.) 33 (Autothrottle)
Other autopilot	4-5	Yes	b	4	5
Electronic attitude director		No	b	1	30
Function Class 3 (Area Navigation and Related Functions)					
Supervisor (mode selection, etc., automatic timing)		Yes	b	4	--
Inertial (cf Note 6)	0,4	Yes	a	2	1-25
Radio					
VOR/DME	4-5	No	d	4	5
Alternates (multiple DME, Omega)	4-5	No	d	4	5
Air data	4-5	No	b	4	--
Optional combination (Kalman filter)	2-3	Yes	a	4	1/5
Processing of flight data (enroute)	2-3	No	b	4	5
Airspeed, altitude	2-3	No	b	4	8,16
Display					
Graphics	2-3	No	b	4	1,8
Text	4-5	No	b	4	10
Function Class 4 (Communications, ATC)					
(4)Collision avoidance (cooperative)	1-2	Yes	b	4	1/3,670
(5)Data communications, programs					
Aircraft (internal)	--	Yes	a	--	1/4 to 250
Air/ground/air	--	No	b	4	Up to 4
Function Class 5 (Support Systems)					
Aircraft integrated data system	4-5	No	a	5	1/4 to 4
Instrument monitoring	2-3	Yes	a	4	5
System monitoring and management	2-3	Yes	a,b	1-4	1/2
Life support monitoring and management	3-4	Yes	b	1-4	<1/2
Engine systems control and operation	1-2	Yes	a	1-2	33

TABLE 3

DATA PROCESSING REQUIREMENTS

	Memory		Operations/Second		
	Instructions	Data/Buffers (32 bits)	Shorts	Longs	Comp. Time (percent)
Function Classes 1 and 2 (Attitude and Flight Path Control)					
Digital attitude control (with stability augmentation)	1845	230	8400	4040	9.9
Active flutter control	70	22	17500	4250	27.6
*Active gust maneuver load control	45	15	7200	1680	5.6
Automatic flight path control					
*Automatic landing	750	275	23100	7900	22.0
Other autopilot	150	100	--	--	--
*Electronic attitude director indicator	790	520	46000	7700	30.8
Function Class 3 (Area Navigation and Related Functions)					
Supervisor (mode selection, etc.)	75	15	--	--	--
*Inertial	2100	150	14300	4860	13.5
Radio					
VOR/DME	250	50	1450	600	1.7
Alternates (multiple DME, Omega)	400	105	--	--	--
Air data	110	25	--	--	--
Optimal combination (Kalman filter)	250	65	150	50	0.2
Processed flight data (enroute)	450	100	10000	4400	11.0
*Airspeed, altitude	360	70	3500	1320	3.6
Display					
*Graphics (horizontal situation indicator)	890	5360	16200	3900	12.8
Text	640	8700	15000	1000	7.6
Automatic tuning, flight-leg switching, etc.	200	50	--	--	--
Function Class 4 (Communication, ATC)					
*Collision avoidance	550	600	14900	1570	8.5
Data communications, programs					
*Aircraft (internal)	280	425	7000	--	2.8
*Air/ground/air	550	137	620	--	0.25
Function Class 5 (Support Systems)					
*Aircraft integrated data system	650	650	1200	60	0.6
*Instrument monitoring	1800	100	5800	2040	5.6
*System monitoring and management	900	50	480	170	0.5
*Life support system monitoring and management	900	50	<480	<170	<0.5
*Engine systems control and operation	1300	200	42700	19000	47.5
Totals: Memory	16300	18100	--	--	--
Operations/time--for heaviest load case*	--	--	183500	50400	154.0

NOTES FOR TABLES 2 AND 3

1. An asterisk indicates functions operational during heaviest requirement on computer - the instrument approach and landing phase under conditions of zero visibility (Category III-C).
2. A dash in these tables indicates an innapplicable column heading, or a function with negligible or nonsustained load.
3. Two numbers separated by hyphen indicates a range of possible values. Two numbers separated by a comma indicates a major and a minor iteration cycle.
4. Figures for collision avoidance are based on Time/Frequency CAS. SECANT System could have time requirement four to five times the stated figure. Also, the memory requirements would probably be increased substantially.
5. The data communications time requirements do not include provision for the actual data transfers because simultaneous program and I/O operations are to be implemented in independent memory banks. Estimates are for approximately 6450 and 3700 input and output data transfers per second, respectively.
6. The Inertial System Criticality Class of (2) pertains to trans-oceanic travel where no other navigation and separation-assurance facility is universally available. However, if radio-based systems with world-wide capability are universally introduced, the Criticality Class would become (4). This applies currently to continental operations.
7. Criticality Classes:

    1. Immediate safety of flight impact.
    2. Eventual safety of flight impact.
    3. Significant change-of-mission impact.
    4. Operational impact.
    5. Economic impact.

    Backup Classes:

    a. No backup.

    b. Backup external to computer only.

c. Backup internal to computer only.

d. Both forms of backup

For further description, refer to Section II.

8. The estimates for totals are derived on the assumption that commonly used mathematical functions--e.g., SIN, COS-- and computer management functions--e.g., dispatching, loading--are available within the appropriate elements of the central complex. The computational requirement of these functions is architecture dependent, and hence is considered as part of the architecture design.

or several of them for one function. This type of entry signifies that for the functions indicated several iteration loops operating at different rates are involved in computation of the function.

Thus, the entries in the column marked Computer Time (Percent) represent the percent of the computing time of the reference machine that the indicated function would consume.

No provision for bulk storage of data is indicated in these tables. While some such provision is desirable, we have not investigated the extend of added storage desirable because differences in size of such a store would have little or no significance for the computer architecture design task. Hence, while it does appear that some bulk store capability is desirable the amount required is not relevant for the purposes of this study.

Because many of the functions studied are currently operative only in experimental digital form, or not currently implemented digitally at all, we were unable to define precise requirements for function iteration rates for many of these functions. Future design and experiment programs will have to take place before definitive iteration rates can be determined. Thus, we have used the iteration rates developed from our analyses and best judgment, and have development computational requirements parametrically in terms of these iteration rates and the number of "shorts" and "longs" to facilitate re-estimation of functional computational requirements, in the event that different iteration rates and/or computation rates prove to be desirable.

IV DISCUSSION OF COMPUTATION REQUIREMENTS ANALYSIS FOR EACH FUNCTION

A. General

In the following pages we document the process analyzed to estimate computation requirements. For each function discussed in this section we indicate the type of information used in our analysis, the technical and operational assumptions made, the character of computation involved for each function, and the approach to estimation. We have discussed simplifications used to facilitate the analysis as well, where appropriate, and have indicated areas where information available to us is incomplete to the extent that confidence in the accuracy of the results obtained is not warranted.

B. Digital Attitude Control Including Conventional (Rigid Body) Stability Augmentation and Handling Qualities Modification, for all Flight Conditions other than Automatic Landing

The computational requirements for this function are derived from an algorithm developed by R. Montgomery of NASA Langley Research Center.[1*] Since the parameter values used in the reference paper are for a different aerodynamic design than the ATT, we have assumed as a preliminary estimate an iteration rate of 20 per second for the attitude control loop. While this rate is apparently adequate for use in an enroute environment, the rate needs to be higher for the approach and landing phase.[†] A rate of 160 iterations per second is used in conjunction with automatic landing (see IVD).

We assume for our analysis the following control surfaces: Rudder, tail elevator, paired ailerons or flaperons, and an individually controllable pair of inboard spoilers, neglecting trim surface, flaps, and any aero-elastic mode control surfaces (these later are treated in Section IVC). This gives five control commands to be computed each 0.05 second.

The following notation is defined for purposes of stating the control algorithm:

[*] References are listed at the end of the report.

[†] Discussions with H. Tobie, the Boeing Company, Seattle, Washington

Let: μ be the control (5) vector at time k

θ be the "staleness" of the sensor information at control application (should be 0.02 second or less here)

X_k be the aircraft state vector at k (assume six translation and six rotational components)

τ be the control iteration rate (assumed 0.05 second here).

Then the computation is, assuming constant τ and θ,

$$\mu_k = K\mu_{k-1} + HX_k + E\mu p_k,$$

where μp_k is the pilot input, a three-vector, K is 5 x 5, H is 5 x 12, E is 5 x 3.

The components of the matrices K, H, and E are assumed programmed over four ranges of velocity and three of configuration (two of these being in the low-speed range only). This gives six set of matrices.

Since constant τ and θ appears satisfactory, there is no need to be concerned with the derivation of the matrices, which appears in the referenced paper.

The state vector may be assumed to be smoothed by some estimation algorithm (e.g., Kalman filter), based on sensor measurements. The computational aspects of this process are estimated as that derived in SRI Project 8274 (Flight Control Data Instrumentation Internal Redundancy).[2]

The assumed filter update rate is 5/second. Because the resulting state estimate is derived from past as well as current information, it must be protected from loss in the case of computational failure.

The output of any autopilot function (other than autothrottle) is assumed to feed into the attitude control computation in lieu of the pilot command input term $E\mu p_k$.

The numerical values for the computation requirements based on the equations described above are summarized in Table 3 of Section III.

C. Active Flutter Control and Gust/Maneuver Load Control

This section describes the analysis of stability and flutter control. Equations developed by Boeing are used for the analysis. Input and output parameters are described, and the transforming functions are presented in a form suitable for use in a sampled data system using a digital computer.

Since these functions are not currently a part of civil aviation aircraft design practice, the source equations and their subsequent transformations should be considered only as examples of what might be encountered.

1. Stability Augmentation

We use the model of a gust/maneuver load stability augmentation system as developed by Boeing.[3] Their concern was to control the vertical and lateral linear accelerations in the passenger compartment. From 727 and 720 B performance, and from the results of SST simulator tests, there were established criteria for the maximum g forces in the cabin, i.e., 0.11 g vertical and 0.055 g lateral. The source of the g forces was turbulence, arising from wind gusts. There exists a model for the statistical distribution of the rms gust velocity as a function of altitude. By applying several simplifying criteria, the gusts of interest were reduced to 5.6, 8.2 and 9.8 ft/s rms, for cruise, descent, and landing approach conditions, respectively.

Boeing designed two augmentation systems to maintain the passenger compartment accelerations within the prescribed limits. For vertical acceleration control there is sensed:

(1) The vertical acceleration at the aircraft's center of gravity; control signals are developed for driving the aft segment of the full span trailing edge flap.

(2) The pitch angular rate; control signals are developed for the elevator.

For lateral acceleration control, the rudder is controlled by signals that measure aft body lateral acceleration, and yaw rate.

The Laplace transforms of the surface control signals are given as follows:

$$F_S(S) = K_1 \frac{S}{S + 0.1} \ddot{X}_S(S) \qquad (1)$$

$$E_S(S) = C_{ES}(S) + K_2 \frac{2}{S + 2} \dot{\theta}_S(S) \qquad (2)$$

$$R_S(S) = C_{RS}(S) - K_3 \frac{S}{S + 0.15} \ddot{Y}_S(S) + K_4 \frac{S}{S + 0.25} \dot{\psi}_S(S), \qquad (3)$$

where the indicated transforms apply as follows:

$F_S(S)$ → flap control

$E_S(S)$ → elevator control

$R_S(S)$ → rudder control

$\ddot{X}_S(S)$ → acceleration at the center of gravity

$C_{ES}(S)$→ pilot's elevator command

$\dot{\theta}_S(S)$ → pitch angular rate

$C_{RS}(S)$→ pilot's rudder command

$\ddot{Y}_S(S)$ → aft body acceleration

$\dot{\psi}_S(S)$ → yaw rate

K_1, K_2, K_3, K_4 Constants during any given flight condition (cruise, descent, landing approach) but differ from condition to condition.

Where the control system is implemented by means of a digital computer, it is more appropriate to use sampled data system concepts, and the Z transform. We thus recast the set of three equations into Z transforms, as shown below:

$$F_Z(Z) = K_1 \left(\frac{1}{1 - e^{-T}Z^{-1}}\right) Z\left(\frac{d}{dt} X(\ddot{t})\right) \tag{4}$$

$$E_Z(Z) = C_{EZ}(Z) + K_2 \left(\frac{2}{1 - e^{-2T}Z^{-1}}\right) \dot{\theta}_Z(Z) \tag{5}$$

$$R_Z(Z) = C_{RZ}(Z) - K_3 \left(\frac{1}{1 - e^{-0.15T}Z^{-1}}\right) Z\left(\frac{d}{dt} \ddot{y}(t)\right) \tag{6}$$

$$+ K_4 \left(\frac{1}{1 - e^{-0.25T}Z^{-1}}\right) Z\left(\frac{d}{dt} \dot{\psi}(t)\right) ,$$

where T is the time interval between samples.

If f(t) is a continuous function in t, its sampled form will be $f^*(t)$, where

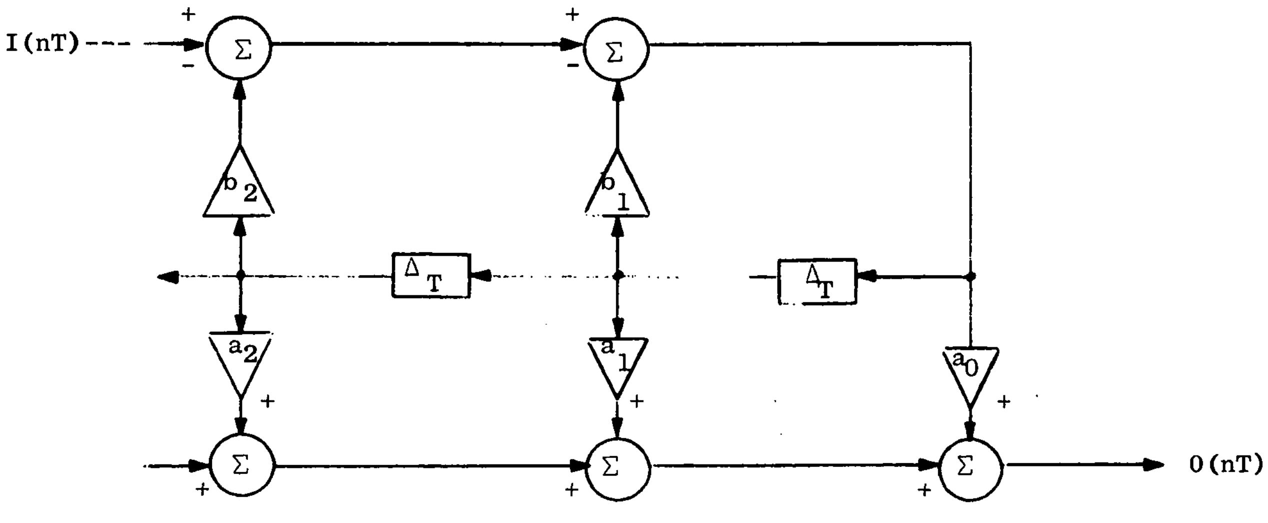
$$f^*(t) = f(t) \sum_{n=-\infty}^{\infty} \delta(t - nT).$$

and $\delta(t - nT)$ represents an impulse of unit area at a time nT. Equivalently, we obtain

$$f^*(t) = f(nT) \sum_{n=-\infty}^{\infty} \delta(t - nT).$$

In the same manner, we can treat the other input and output functions: $e(t)$, $r(t)$, $\ddot{x}(t)$, $c_E(t)$, $\dot{\theta}(t)$, $c_R(t)$, $\ddot{y}(t)$, and $\dot{\psi}(t)$.

The transforming function is treated as being implemented by a system as shown below:



where Δ_T is a delay of time T.

For Eq. (4), the filter function is $\dfrac{1}{1 - e^{-T}Z^{-1}}$.

If i(nT) is the value of the nth sample to the filter, we derive the filter system to be

$$f(nT) = i(nT) + e^{-T}\, f((n - 1)T), \text{ with}$$

$$i(nT) = K_1\ ((\ddot{x}(nT) - \ddot{x}\ ((n - 1)T))/T) \text{ so that}$$

$$f(nT) = \frac{K_1}{T}\ (\ddot{x}(nT) - \ddot{x}\ ((n - 1)T)) + e^{-T} f((n - 1)T). \tag{7}$$

In an analogous form, we develop that

$$e(nT) = c_E(nT) + 2K_2(\theta(nT) + e^{-2T}\ \bar{e}((n - 1)T)), \tag{8}$$

where $\bar{e}\ ((n - 1)T) = \theta((n - 1)T) + e^{-2T}\ \bar{e}\ ((n - 2)T))$.

The equivalent for Eq. (6) is given to be

$$r(nT) = c_R(nT) - \bar{r}(nT) + \bar{\bar{r}}(nT), \tag{9}$$

where $\bar{r}(nT) = \dfrac{K_3}{T}\ (\ddot{y}(nT) - \ddot{y}((n - 1)T)) + e^{-0.15T}\ \bar{r}\ ((n - 1)T)$

and $\bar{\bar{r}}(nT) = \dfrac{K_4}{T}\ (\dot{\psi}(nT) - \dot{\psi}((n - 1)T)) + e^{-0.25T}\ \bar{\bar{r}}((n - 1)T)$.

2. Flutter Control

The flutter control system has as its objective the control of vertical and torsional oscillation modes of the wings. The control system used here as a model was described by Boeing.[4]

The particular concept of interest to us provides control signals only to the outboard trailing edge surface of the wing. No control is provided of other postulated wing surfaces, i.e., leading and trailing edges of the inboard and mid-span surfaces, or the outboard leading edge surface.

The equations used in the Boeing analysis were referred to the wing chord, 240 inches long, that passed through the outboard surfaces.

Each surface, leading and trailing, was 0.2 chord wide. Vertical acceleration was sensed at a 0.3 chord point, and a 0.7 chord point.

The signal for controlling the trailing edge surface is $\epsilon(t)$:

$$\epsilon(t) = \frac{2G(2,1)}{c} \int \frac{\int\int \ddot{h}_1(t)dt}{w_1(t)} dt + \frac{5C(2,2)}{4} \int \int (\ddot{h}_2(t) - \ddot{h}_1(t))dt\ dt$$

$$+ \frac{5G(2,2)}{4} \int \frac{\int\int(\ddot{h}_2(t) - \ddot{h}_1(t))dt}{w_{21}(t)} dt \quad , \tag{10}$$

where

$$w_1(t) = +\sqrt{\left| \frac{\ddot{h}_1(t)}{\int\int \ddot{h}_1(t)dt\ dt} \right|} \tag{11}$$

and

$$w_{21}(t) = +\sqrt{\left| \frac{\ddot{h}_2(t) - \ddot{h}_1(t)}{\int\int(\ddot{h}_2(t) - \ddot{h}_1(t))dt\ dt} \right|} \quad . \tag{12}$$

The other variables and constants are defined below:

$\ddot{h}_1(t)$ = the acceleration measured nearest the leading edge

$\ddot{h}_2(t)$ = the acceleration measured nearest the trailing edge

c = the chord length

$$\left.\begin{matrix} G(2,1) \\ G(2,2) \\ c(2,2) \end{matrix}\right\} = \text{system constants.}$$

An essential ingredient of the control concept is the use of a rate signal divided by frequency. It is apparently assumed that the measured accelerations are dominated by sinusoidal time functions so that the notion of frequency has meaning. Given the sinusoid, then Eqs. (11) and (12) can be used to obtain instantaneous frequency. Alternatively, zero-crossings can be detected and the period measured. Boeing implemented both techniques using analog elements. Problems of noise appeared, causing w(t) to become 0 at times, thus upsetting the integration process.

Protection against these occurrences is not provided here.

If the input signals $\ddot{h}_1(t)$ and $\ddot{h}_2(t)$ are sampled at a rate of $\frac{1}{T}$, then we represent the sampled signal at the nth sample time as $\ddot{h}_1(nT)$ and $\ddot{h}_2(nT)$, respectively. By integration of the samples we obtain

$$\dot{h}_1(nT) = \ddot{h}_1(nT) + \dot{h}_1((n-1)T)$$

$$h_1(nT) = \dot{h}_1(nT) + h_1((n-1)T)$$

and similarly for $\ddot{h}_2(t) - \ddot{h}_1(t)$.

Using these sampled inputs, we develop Eq. (13), which incorporates the features of Eqs. (10), (11), and (12):

$$\varepsilon(nT) = a_1(nT) + a_2(nT) + a_3(nT), \tag{13}$$

where

$$a_1(nT) = K_{11}\dot{h}_1(nT)\ p_1(nT) \tag{14}$$

$$\dot{h}_1(nT) = \ddot{h}_1(nT) + \dot{h}_1((n-1)T)$$

$$p_1(nT) = +\sqrt{\left|\frac{h_1(nT)}{\ddot{h}_1(nT)}\right|}$$

$$h_1(nT) = \dot{h}_1(nT) + h_1((n-1)T)$$

$$a_2(nT) = K_{12}\ \Delta h(nT) \tag{15}$$

$$\Delta h(nT) = \Delta\dot{h}(nT) + \Delta h((n-1)T)$$

$$\Delta\dot{h}(nT) = \Delta\ddot{h}(nT) + \Delta\ddot{h}((n-1)T)$$

$$\Delta\ddot{h}(nT) = \ddot{h}_2(nT) - \ddot{h}_1(nT)$$

$$a_3(nT) = K_{13}\ \Delta\dot{h}(nT)\ p_2(nT) \tag{16}$$

$$p_2(nT) = +\sqrt{\left|\frac{\Delta h(nT)}{\Delta\ddot{h}(nT)}\right|}$$

3. <u>Sampling Rate</u>

We estimate that 240 samples per second will need to be processed for the vertical and lateral stability augmentation systems described by Eqs. (1) through (12). This estimate is derived from a visual examination of some graphs that accompanied the Boeing report. It appeared that the g forces had a maximum frequency component of 24Hz. A minimum of 48 samples per second is thus indicated, with 240 samples being five times that minimum.

The flutter control system had a "frequency range of interest" of 5 to 25 Hz; thus we talk in terms of 250 samples per second there, for each accelerometer pair on each wing.

Equations (8) and (9) form the basis for the stability augmentation estimate, while Eqs. (13) through (16) are used for the flutter control estimate. These are all summarized in Table 4:

TABLE 4

DATA PROCESSING REQUIREMENTS
FOR
STABILITY AUGMENTATION AND FLUTTER CONTROL

		Application	
		Stability Augmentation	Flutter Control
No. of instructions executed per iteration	Shorts	30	70
	Longs	7	17
CPU time required per iteration (microseconds)		232	1104*
Number of iterations per second		240	250
CPU time (in milliseconds) required per second		55.7	276
Program storage required (words)		43	69*
Data storage required (words)		13	22*

*See discussion below.

Driving the stability augmentation programs are five input variables, while two output variables are generated to drive the control surfaces. In making the estimates shown in Table 4, it is assumed that:

- All the inputs are sampled each iteration by the execution of a single input instruction, while both outputs are transmitted each iteration by the execution of a single output instruction.
- The input and output instructions were "short," and each required one word of program storage.
- All I/O transfers required no CPU time.
- The transfers were into and out of the data storage area.

The flutter control program was similarly treated, but with four input variables and two output variables.

Flutter control is applied separately to each of the two wings, but a common program serves to control both. Thus only one copy of the code, 69 words, need be kept in storage, but two separate sets of data storage (11 words each, 22 words total) are maintained. One iteration is considered to serve both wings requiring the 1104 microseconds; thus each wing requires only 552 microseconds per iteration.

D. Automatic Flight Path Control

There are two functions in this category: (1) the automatic landing functions and (2) other autopilot functions to be used in enroute mode.

The automatic landing functions are discussed at some length in the following paragraphs. The "other autopilot" functions are not discussed here, however, because most of the functional elements usually considered as the "autopilot" are in this report contained in the navigation* and digital attitude control functions. Only a small amount of memory, for instructions and work data, is required to transfer the necessary navigation-

*For example, computations of bearing (and time/distance) to waypoint, track, cross-track deviation, wind/drift, etc.

related calculations to the digital attitude control command input. The assumed numbers are indicated in Table 3 of Section III. The minimal required computational operations attendant to these transfers, assumed to occur at a rate of 5/second, are included in the time estimates for the navigation and attitude control functions.

1. General Considerations

Particularly with the advent of the larger, more expensive transports such as the 747, DC-10, and L-1011, an all-weather capability becomes increasingly important. The costs of a delayed or diverted flight are very expensive in terms of passenger man-hours, cost of nonoptimum aircraft usage and so on. The most critical factor in flight-schedule reliability is the all-weather approach, landing and roll-out capability (Category III-C). Such a capability will certainly require a fully automatic integration of radio and inertially derived data into the aircraft control system with a sophisticated display system to optimize the pilot monitoring function (see Section IVE, "Electronic Attitude Director Indicator").

Automatic landing systems based on analog processing techniques have been carried to advanced stages of development--in fact, all 747s are so equipped, and are flown extensively in the automatic landing mode in commercial operations under clear-weather conditions.* However, digital systems are just now being subjected to significant research and development. †

*To date, no U.S. airports have been commissioned for fully blind landings (Category III). However, commissioning of the Heathrow Airport at London for a Category III type of operation has recently been reported. BOAC has procured a special Boeing 747 equipped with triplicate hydro-electric actuators for control surfaces for use in connection with flight-test programs. The French have commissioned Caravelle aircraft with analog equipment for Category III-A and -B approaches since 1964. Lear Siegler is connected with the French company responsible for the equipment and reports successful performance with new units designed for the L-1011 that assures Category III-C capability.

†As of February 1972, Boeing had just completed the flight-test portion of an Autoland project for the Federal Aviation Agency (FAA), and was in the process of writing the final report. Lear Siegler reports that it has a current project for development of a digital system for the U.S. Navy.

Definitive design information on such systems is therefore not generally available, and the data-processing requirements data of this report must therefore be considered to be approximate.

Fully-effective automatic landing systems require integration (complementary filtering) of radio and inertially derived guidance data. There are two prime reasons for this:

(1) Radio-based Instrument Landing Systems (ILS) guidance data are subject to spurious "bends" of significant magnitude--due to reflections from both fixed objects and moving aircraft. The short-term stability of the intertial system suppresses these spurious bends.

(2) In the event of the complete loss of ILS ground-based transmissions during the critical final stages of the approach, it would be desirable to avoid aborting the landing. Boeing claims that with ILS failure at any time within 15 seconds of touchdown, automatic landing can be continued on inertial data alone more safely than execution of a "Missed Approach."

The ILS type of system thus far used in automatic landing tests has been the conventional system that constrains the final portion of the approach to two discrete "beams"--the lateral-guidance "localizer" and the vertical-guidance "glide slope." However, for the long term, implementation of a more flexible system of scanning beams providing both lateral and vertical position information throughout relatively wide sectors is virtually assured--the alternatives being investigated are based on microwave transmissions and are designated as Microwave Landing Systems (MLS).* Coupled with precision Distance Measuring Equipment (DME)†

*See Ref. 5. The basic MLS document is Ref. 6. As of January 1972, the FAA has awarded contracts to six firms for first-phase studies on various MLS techniques.

†Projections are for an accuracy of the order of 20 feet.

installed at the projected touchdown point, the aircraft position is to be available in all three dimensions, thereby providing the basis for:

- More flexible specification of approach paths in high-density terminal areas.
- More flexible specification of the vertical profile of the approach to facilitate noise abatement.

However, the MLS specification is restrictive in the sense that vertical and lateral guidance data are to be available only five times per second, and random quantization errors will be involved with all such data. This is in contrast to the conventional ILS system, where the glide slope and localizer data (for the single approach path) are continuously available for sampling at any rate and precision that may be considered appropriate. Boeing reports that data sampling and complementary filtering rates as high as 160 per second are used in the current experimental system in order to achieve the desired high-gain control loops.

In the absence of any information whatsoever on possible automatic-landing processing/control techniques with MLS, the processing requirements derived here are referenced to the finite but limited information available on ILS-based systems. In any event, such systems will be in use for five or perhaps ten years as the MLS systems are designed, tested, and placed into full-scale use. However, it is to be noted that automatic landing in an MLS-based system might require somewhat more computer time and memory for generation (and smoothing) of the specified approach path.

The data-processing requirements summarized below in Section B are based on the following prime assumptions:

(1) For vertical-guidance processing, an iteration rate of 160/second is assumed* (based on recent Boeing work).

*In the absence of definitive design and performance information, this iteration rate appears to be conservatively high. It is possible that this rate may be required in part because of current limitations elsewhere in the initial experimental system, and that the ultimate requirements may be reduced to 100/second or perhaps even lower. In the meantime, the use of the 160/second rate in this analysis contributed to conservative system scaling.

(2) For horizontal-guidance processing, an iteration rate of 20/second is assumed (based on Lear Siegler analog implementation of an experimental system tested for NASA and FAA).[7] This involves computation of inertially derived ground-velocity components at a rate higher than that normally programmed in inertial systems.

(3) For engine control, an iteration rate of 33/second is assumed.

(4) A vertical noise-abatement profile within an initial 6-degree path intercepting a 2-degree glide slope at approximate altitude and range of 400 feet and 1.5 miles is assumed (as per NASA Langley and Ames flight tests).

(5) All input data are presented to the computer in standardized digital code by the various receiver and transducer units (e.g., ILS, DME, radar altimeter, and pitch-, roll-, and yaw-rate units).

2. Summary of Analysis (Automatic Landing)

a. Memory Requirements (not including subroutines)

●	Program	650
●	Data and Buffers (Words at 32 bits each)	275
●	Supplementary (Program)	100

b. Number of I/O Words per Second

●	Inputs	820
●	Outputs	260

c. Processing Operations per Second

		No./Second	Computer Time
●	Shorts	21,400	8.6%
●	Longs	7,900	12.7%
●	Supplementary (Shorts)	1,700	0.7%
	TOTAL		22.0%

E. The Electronic Attitude Director Indicator (EADI)

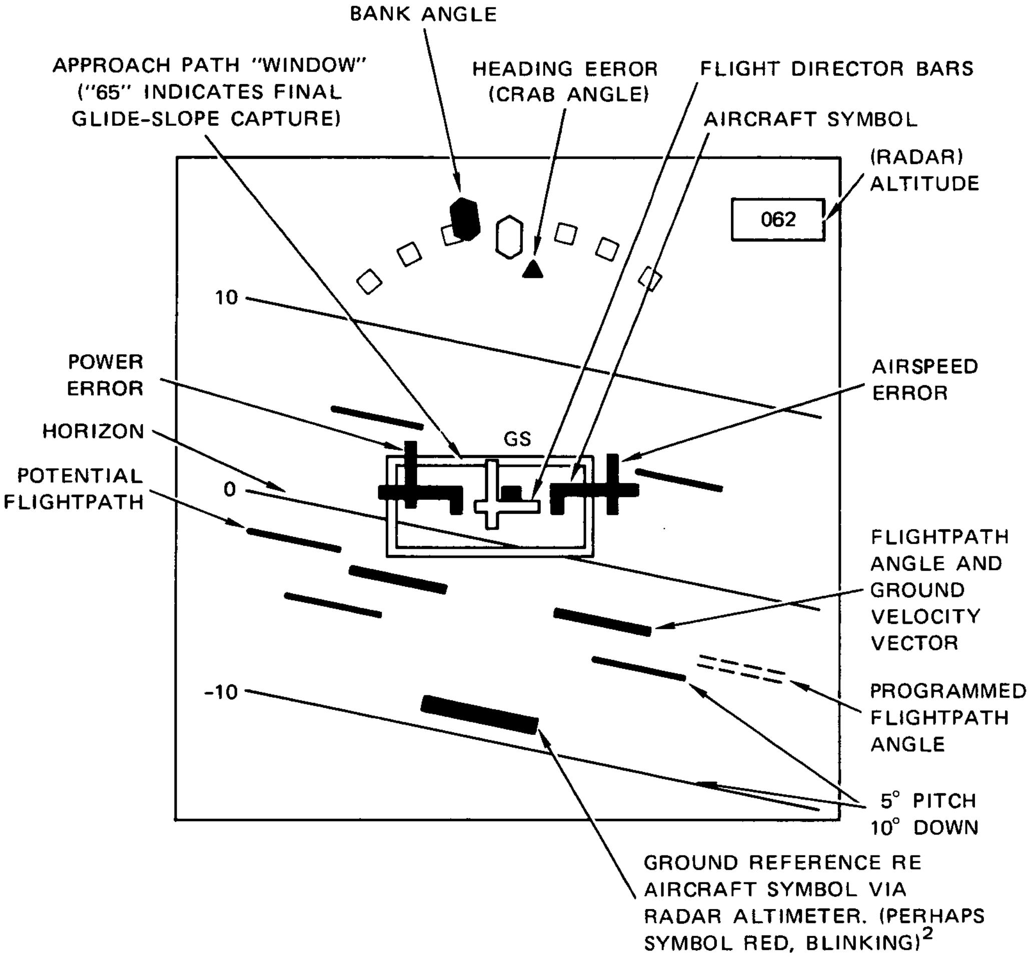
1. General Considerations

As aircraft become larger, and flight operations more sophisticated, there is an increasing need for more effective instrumentation of aircraft attitude, guidance, performance, etc. Research programs on two basic types of Electronic Attitude Director Indicators (EADIs) directed toward fulfillment of future needs have been in progress for some time:

- Cathode-ray-tube (CRT) displays[8,9]
- "Head-up" displays[10]

The Head-up display offers the major advantage that pilot effectiveness is potentially enhanced at the abrupt transition from instrument approach to visual landing conditions because of his uninterrupted "infinity-focus" visual orientation in line with the windshield. However, there are negative factors related to reduced visibility range (particularly under minimal-visibility conditions at the destination airport area), and to restricted flexibility relative to that of a CRT-based system. Accordingly, a CRT-type of system is used as the basis for the analysis of this section of the report. The critical approach/phase of flight operations (corresponding to Autoland) is used in estimation of the data-processing time requirements. Estimations of memory requirements for other possible modes of flight operations in addition to those of the approach/landing mode are also included.

A conceptual EADI display for the approach/landing mode of flight is sketched in Figure 1. The number of display elements typically visualized for even a sophisticated approach mode of operation is not high--in fact, it is essential that clutter be minimized to enhance rapid, error-free interpretation by the pilots. Ideally, a multiple-color display (tube) will be used. This involves data processing to the minimal extent of affixing a simple color code to each element to be displayed. In any event, the common technique of selective symbol blinking will be a powerful technique for directing priority attention to symbols to those parameters


BANK ANGLE
APPROACH PATH "WINDOW" ("65" INDICATES FINAL GLIDE-SLOPE CAPTURE)
HEADING EEROR (CRAB ANGLE)
FLIGHT DIRECTOR BARS
AIRCRAFT SYMBOL
(RADAR) ALTITUDE
062
10
POWER ERROR
HORIZON
POTENTIAL FLIGHTPATH
0
GS
AIRSPEED ERROR
FLIGHTPATH ANGLE AND GROUND VELOCITY VECTOR
-10
PROGRAMMED FLIGHTPATH ANGLE
5° PITCH
10° DOWN
GROUND REFERENCE RE AIRCRAFT SYMBOL VIA RADAR ALTIMETER. (PERHAPS SYMBOL RED, BLINKING)2

NOTES:
1. These EADI Concepts are based primarily on Boeing's current 747 instrumentation and experimental EADI units.
2. The use of multiple colors and selective symbol blinking is desirable for this type of display. As an alternative, the use of dotted and dashed lines (with symbol blinking) would suffice to enhance clarity of presentation.

SA-1406-26

FIGURE 1 CONCEPTUAL EADI DISPLAY

for which a critical (or marginal) situation exists. The CRT display system also provides the capability of superimposing a televised approach or ground-roll image to the EADI presentation, but this feature will not be considered in this report.* It is basically unrelated to the data-processing system.

For an intensive-use application such as the EADI, it is essential that the display be of high definition and that it be free of any flicker or "jumpiness" of the symbols.† The first two factors are related primarily to the display hardware (except for the requirement that the computer supply coordinate information with enough bits to fulfill definition requirements). The "jumpiness" factor appears to require a full computation and update rate of the order of 30/second.‡ Even at this rate, there will undoubtedly be some jumpiness when the approach path symbol moves toward the aircraft symbol as the flight path approaches the specified vertical profile. However, this is but a momentary high-speed movement and can probably be accommodated--perhaps even masked out until near-capture of the approach path. Another movement-critical symbol is that representing ground reference, this symbol being programmed for display throughout the final critical descent just prior to touchdown. It graphically indicates relative radar-derived altitude above ground for perhaps the final 75 feet, touching the aircraft symbol at touchdown. At typical descent rates, the symbol might move at a rate of approximately 20 mils per iteration (at 30/second).

---

*Because of the requirements for maximum symbol fidelity, this evaluation is based on the assumption of stroke-generated rather than roster-generated displays. This would complicate incorporation of the televised image feature.

†Ref. 8

‡This is in contrast to the Graphic Display Unit in the horizontal map mode, where movements of all display elements are consistent and relatively slow. For this situation, a full computation only once per second will be adequate, intervening movements being accommodated by simple translation and rotation of the entire display approximately eight times per second.

The estimate of computer time requirement summarized below is relatively high, the major requirement being associated with those symbols subject to rotation.* If it ultimately develops that this magnitude of requirement cannot be accommodated, it is possible that significant gains could be realized by one or more of the following factors:

- Decrease in the update rates of at least some of the symbols.
- Use of display equipment providing a hardware capability for symbol display with both translation and rotation (symbol sets requiring different translation and rotation would necessarily be programmed and activated separately).
- Use of special computer hardware specifically designed for maximum efficiency for coordinate rotation and translation.

*To minimize time requirements for generation of these symbols, only one point per end of each bar segment (line) is specified and computed--the width of the bar being generated by addition of a hardware-generated circular deflection component to the linear deflection. The intensity of a bar generated in this manner will not be perfectly uniform over the entire area of the bar, but it should be satisfactory, maximum intensity being along the edges.

2. Summary of Analysis (Electronic Attitude Director Indicator)

a. Memory Requirements (not including subroutines)

● Program (instructions)	690
● Buffer/Data (words at 32 bits each)	520
● Supplementary (Program)	100

b. Number of I/O Words per Second

● Inputs*	0
● Outputs	175

c. Processing Operations per Second

	No./Second	Computer Time†
● Shorts	44,000	17.6%
● Longs	7,700	12.4%
● Supplemental (Shorts)	2,000	0.8%
Total		30.8%

The following subjects are discussed in this section: Inertial Navigation, Area Navigation, Navigation Graphics Display, and Display of Navigation-Related Text.

1. Area Navigation Systems (RNAV)

a. General Considerations

In the recent past, continental navigation has been performed largely on the basis of routes established by Very-High Frequency Omniranges (VOR) and Distance-Measuring Equipment (DME), both ground-based radio systems. More recently, the introduction of new navigational capabilities is bringing about a number of improvements being embodied in "Area Navigation Systems" (RNAV).[11] The new factors making this possible are inertial navigation systems first introduced

*The required inputs to EADI will already have been placed in memory in connection with the requirements of other functions.

†The computer time requirements are derived for the (worst-case) approach/landing mode of flight.

via the Boeing 747 fleets*, and on-board digital data-processing capability. With these facilities, the following new capabilities become available (among others):

- A "blending" of the long-term stability of radio-based navigations systems with the short-term stability of the inertial systems, thereby achieving a substantial improvement in accuracy throughout the entire flight--blending to be achieved via optimal filtering (e.g., Kalman).
- The ability to use routes not only directly along a sequence of fixed points coinciding with ground-based radio transmitting locations, but along any desirable offset route, thereby reducing traffic congestion and its potential collision dangers, etc.
- Automatic advance planning and postfailure analysis to determine and implement the best possible navigational capability. For example, in the event of failure of one radio-based source of navigational information, data acquisition and computation can be shifted to other sources of the same or different type.† Also, in the event of failure of the inertial systems, airspeed, heading, and wind-vector data can be substituted as a backup for the inertially derived inputs.
- Computation and display of navigation-related information such as distance and time to waypoints along the route, cross-track deviation, true ground speed and track, wind vectors, etc.

*Triplicate inertial systems were installed as standard equipment in each Boeing 747, Inertial systems are considered in Section of this report.

†It is possible that radio-based navigation techniques will be changed significantly within the next few years. For example, "Omega," a very-low-frequency hyperbolic system is currently being proposed as the prime world-wide navigation system (see Ref. 15).

- Storage and processing of enroute and terminal navigation-facility data for automation of flight-plan organization and display, automatic route-leg switching and radio tuning, etc.

A number of companies are currently developing and testing RNAV systems, and it is anticipated that their systems will become widely operational within the next two or three years.

Except for the inertial and the display systems that are considered separately in other sections of this report, the requirements of the various RNAV components are summarized below and tabulated in more detail in Appendix B.

b. Summary of Analysis (RNAV)

1) Memory Requirements (Not Including Subroutines)

● Program (Instructions)	2,100
● Data and Buffers (Words at 32 bits each)	480

2) Number of I/O Words per Second

● Inputs	90
● Outputs	120

3) Processing Operations per Second

	No./Second	Computer Time
● Shorts	15,100	6.1%
● Longs	6,400	10.4%
Total		16.5%

2. Inertial Navigation Systems

a. General Considerations

Inertial navigation systems--even in their relatively short life span of approximately three years--are revolutionizing air navigation. As completely stand-alone systems independent of all other airborne and ground-based equipment, they are providing the aircraft operator with high levels of operational flexibility and self-sufficiency.

Inertial equipment is the heart of the navigation system on all 747 aircraft, triplicate units being installed as standard equipment. This redundancy not only provides the operational assurance against a unit failure but also provides a built-in contribution to economic feasibility by eliminating the requirement for an aircraft navigator. At estimates based on 5-navigator man-years per aircraft year, this corresponds to an annual saving in salaries alone of approximately $125,000. At approximately $60,000 per unit inertial system, the equipment is amortized in approximately two years.

Even in their relatively short life span of approximately three years in commercial operations, the inertial systems have found full operational acceptance from the airlines--not only for transoceanic flights but also for domestic flights. In fact, the chief engineer of one U.S. airline has stated that attempts of the company to remove only one of the three inertial units from aircraft involved in only domestic operations met with so much opposition from the pilots that all three units were retained.

The inertial units provide a wide range of information both to the pilot and to aircraft systems. The major parameters are as follows:

- Latitude/longitude
- Distance/time to next "waypoint." Provision is made for accommodating as many as nine waypoints along a route at any time.[12]
- Cross track distance/track angle error
- Ground speed
- Heading/drift
- Track
- Wind
- Aircraft attitude

As currently constituted, each of the inertial units operates on a completely independent basis--even to the extent of including its own battery to provide uninterrupted operation throughout interruption of primary power (for perhaps 15 minutes maximum). The data of each unit are available to the pilots on individual displays, and are accessed as appropriate to the rest of the system for:

- Basic Navigation--Processing with radio-derived data to derive minimum-error position. This technique combines the superior short-term accuracy of the inertial system with the less precise, but long-term-stable radio-derived data.
- Altitude Information--Processing with barometric instruments to derive altitude and altitude-rates, information with both short-term and long-term stability.
- Aircraft Control--Display of aircraft attitude, and supply of data for computation of "Flight-Detector" command signals.
- "Autoland" Operations--Processing with Instrument Landing System (ILS)--derived data to provide maximum accuracy and stability throughout the critical approach landing and rollout phases of flight. In addition, it is claimed (by Boeing) that even in the event of complete loss of ILS information within 15 seconds of touchdown, the inertial system can provide sufficiently accurate guidance to touchdown, this being safer than execution of a "Missed Approach."

Some of the specific considerations that influenced the analysis of the data-processing requirements of the inertial systems are as follows:

(1) Polar Capability--Some inertial system configurations preclude operation at a latitude (λ) beyond approximately 70° because of intractable computations and

gyro orientations. Because of the global nature of current and projected airline operations, it is essential that the system used as a basis for analysis in this report be operable at any latitude.* A separate set of equations is required for the final portion of the computations for near-polar latitudes. A moderate amount of program core-storage is required to accommodate this program-shift situation, but there is no problem in regard to time. Even if more time were required for the polar computations, the computer could handle them because of a relatively low total workload while cruising in such regions.

(2) Special Hardware--It is assumed that short-term algebraic integration of the positive and negative pulses generated by the accelerometers is implemented by special hardware "up/down" counters associated individually with each of those components. The components (approximately 100 to 120 bits) of these registers are strobed into memory periodically by the computer as the basic digital inputs to the navigation computations.† The strobe rate is assumed to be 25/second (Collins uses this rate; Delco uses 20/second; for a system, Bendix uses 50/second).

(3) "Strap-down" Systems--Techniques based on using inertial components "strapped" directly to the vehicle frame are being investigated by the industry, the prime goal being the elimination of the need for a stable

*The system on which the analysis of this section is based is System E of Ref. 13.

†Collins performs these integrations with differential analyzers (DDAs). Special "CORDIC" computer units are used for processing some of the spherical trigonometric computations, data conversions, etc. (See Ref. 14.).

table. However, such a system subjects these inertial units to the high angular rates of the vehicle (of the order of a radian/second), still demanding the same accuracy as that required of inertial components isolated on the stable table of conventional systems. This constitutes a significant problem. Also, spurious factors such as cross-coupling that do not incur significant errors in the gimbaled system will most probably constitute a real problem in "strap-down" systems. There are no known strap-down systems being planned for aircraft navigation use at the present time, so no consideration was given to such techniques. (However, Bendix has used a strap-down system in a nonaircraft guidance application.)

(4) Timing--Very accurate timing is required for initiating the basic sampling/integration functions of the navigation computations. It is assumed that oscillator/divider hardware performs the actual timing function--this generating interrupts at the appropriate intervals for initiation of the various classes of the computer processing.

(5) Computation Rates--The "inner-loop" integration is assumed to be implemented at a rate of 25 per second. It is further assumed that the basic navigation computations are performed at a rate of 5 per second, this corresponding to a traveled distance of approximately 200 feet at Mach I. The display-related processing rate is assumed to be only 2 per second.

b. Summary of Analysis (Inertial Systems)

Detail on the analysis of the data-processing requirements of an inertial system is provided as one of the illustrative examples in Appendix B. The figures there and as summarized below are for a single inertial system.

1) Memory Requirements (Not Including Subroutines)

● Program (Instructions)	1800
● Buffer/Data (Words at 32 bits each)	150
● Supplementary (Program)	300

2) Number of I/O Words per Second

● Inputs	201
● Outputs	234

3) Processing Operations per Second*

	No./Second	Computer Time†
● Shorts	12,323	4.9%
● Longs	4,860	7.8%
● Supplemental (Shorts)	2,000	0.8%
Total		13.5%

3. Graphic Display Unit (GDU)

a. General Considerations

The data-processing requirements of a Control Display Unit (CDU) for providing a bi-directional interface between man, the aircraft system, and the ground-based system are analyzed in Section IVF-4. In the CDU, all data are processed and displayed in text form. This section deals with an additional type of display system for presentation of navigation and flight-control information in a graphic rather than text format. Though not yet developed to an advanced stage in typical RNAV systems now being prepared for marketing, this type of graphic display in some form will almost certainly be included in the future because of its effectiveness in rapidly communicating to the pilots both absolute and

*The inertial-system processing rate is probably higher than indicated here during initialization. However, this is a preflight operation at which time computing capacity should be more than adequate.

†The Delco Carousel IV integrates at 20/second. It is a serial unit at approximately 6 and 110 μs for "shorts" and "longs," respectively. Computer time requirements for these figures would be approximately 50 percent total for the functions considered here. This, plus allowances for executive control, self-check, and margins, appears to be reasonable.

relative relationships among a number of complex flight-related parameters.* Also, the type of presentation, the scale, the mode, etc., can be changed rapidly to accommodate the needs of the moment.

As is the case with the companion CDU display system, the analysis here for the GDU is based on the following types of assumptions:

- The computer periodically formats the graphic information (and supplementary alphanumerics), and transmits it to the display unit (GDU) for display. Fundamentally, a point (or line segment of a polygon) is transmitted as X and Y coordinates, and related identification/control discretes are packed into one word (e.g., 32 bits). An iteration rate of once per second is used in this analysis.†
- The GDU hardware provides the facility for functions such as the actual symbol and character generation, symbol blinking, selective intensification, and display refreshing, thereby freeing the computer of any related responsibility except for the appending of simple function codes as appropriate to specify selective blinking or intensification (and perhaps color of display).

It is further assumed that there are to be both horizontal and vertical types of presentation, each with two display modes at four scales each:

*Initial specification for CRT or projection systems are discussed in ARINC Project Paper 588, dated 12 May 1971.

†To prevent a "jerky" appearance of the presentation for those cases where many elements may be moving, simple translation and rotation are assumed to be implemented by the GDU under computer control at a rate of 8/second. This precludes the need for recomputing all elements at this relatively high "update" rate.

(1) Horizontal

- Track orientation (up) with the aircraft fixed position at a point slightly below the center of the display.
- True-north (up) orientation with aircraft position/heading indicated (if the aircraft is within the boundaries).

(2) Vertical

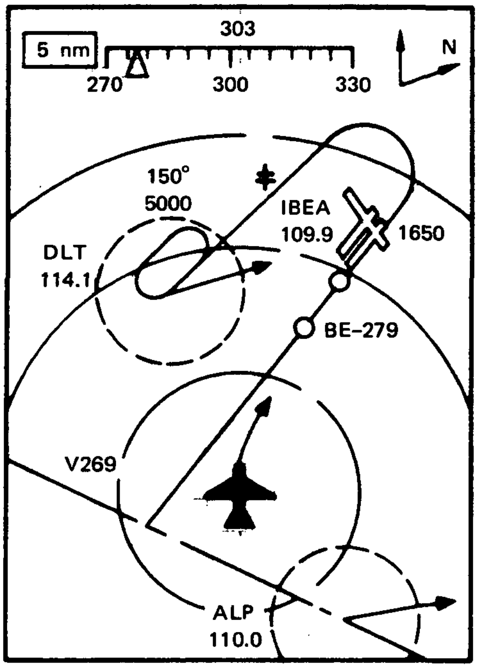
- Profiles on a grid of speed-versus-altitude from sea level to a maximum value (e.g., 50,000 ft), with aircraft "position" and trend vector and "target" indicated.
- Aircraft-centered profile with trend vector and target indicated.

There will be much commonality of symbol and line-drawing techniques and stored data among these various presentation modes. Also, the navigation information concerning the ground-based facilities, route structure, etc., is common to the horizontal display requirements considered here, and to other navigation functions, automatic tuning of navigation and communication units, etc., considered elsewhere. The data-storage requirements of all such functions are included in this section of the report.

The type of display that would be generated for the approach/ landing mode of flight operations is sketched in Figure 2. The track-oriented mode is used here, the Instrument Landing System (ILS), navigation, and airport features being presented in positions related to the fixed aircraft symbol.* A trend vector is computed and displayed to indicate the aircraft's projected flight path.

*The Collision Avoidance System (CAS) potentially uses the same general type of display. It is possible that in practice, the same display hardware could be used for simultaneous presentation of the GDU and the CAS data. Further, it has been suggested that ground-derived information also be displayed on the GDU--for example, locations of weather or turbulence areas, and the positions (and vectors) of aircraft that present potential collision or congestion threats (in addition to those derived via CAS). If these features were to be implemented, the GDU processing load would be increased by as much as perhaps 30 percent.

Estimation of the data-processing requirements of the GDU is based solely upon analysis of this type of horizontal display, because only one mode of presentation can be implemented at any one time, and this mode is the most demanding of computer processing. This is particularly true of the approach and landing mode used here in the analysis because of its relatively heavy demands in terms of the number of displayed points that must be computed relative to aircraft position, and then rotated with reference to aircraft heading in the track-oriented display mode.



SA-1406-27

FIGURE 2 TRACK-ORIENTED GRAPHIC DISPLAY— THE APPROACH/LANDING MODE

NOTE: The path displayed beyond and to the left of the airport symbol specified the Missed Approach Procedure (MAP) to be executed in the event that the landing must be aborted.

b. Summary of Analysis of the Graphic Display Unit (GDU)

Detail on the analysis of the data-processing requirements of a Graphic Display Unit is provided as one of the illustrative examples in Appendix B.

1) Memory Requirements (Not Including Subroutines)

● Program (Instructions)	690 Words
● Buffer/Data (words at 32 bits each)	260 Words
● Base (words at 32 bits each)	
Horizontal Displays	4,500*
Vertical Profiles (Additional)	600
● Supplementary (Program)	200

2) Number of I/O Words per Second

● Inputs	23
● Outputs	160

3) Processing Operations per Second

	No./Second	Comp. Time
● Shorts	13,700	5.5%
● Long	3,900	6.3%
● Supplemental (Shorts)	2,500	1.0%
Total		12.8%

4. Navigation-Related Textual Display

This subsection discusses our estimate of core storage and CPU utilization presented by the display of textual material on CRT terminals. Because this estimate is particularly sensitive to the assumptions behind it, these assumptions are discussed in some detail.

*Comparative Data--Collins estimates a requirement of from 1000 to 1500 words for only the basic data on navigation aids, route structure, etc. Therefore, this 4500-word estimate should not be far out of line for the GDU functions considered here.

As a point of departure for this analysis, we use the "Control and Display Unit" (CDU), of the Collins Radio Company's AINS-70 Area Inertial Navigation System. From that system we obtain significant information regarding the types and amounts of data presented to the aircraft crew, and the desired responsiveness of the display subsystem. Collins, for their application, has estimated the amount of core storage and CPU time required by the CDU in their hardware system. Their estimates are used to support the estimates provided here.

a. The Major Assumptions

1) Display Characteristics

The aircraft will be equipped with four CRT displays, two each for the captain and the first officer. Each display may present information different from all the others, or displays may present identical information. A typical display of text would consist of 16 lines of 16 characters each, where a character may be any of 64 different symbols. By means of push buttons on the console, the user can select any one of 60 different pages.

Each page is distinguished by its format, which dictates what fixed and variable information is to be presented, and the forms of their presentation. The variable information represents, in part, the present or future state of the aircraft, while the fixed information primarily serves to describe the variable information.

2) Display Control

The hardware controller of each display is assumed to contain all the facilities for refreshing the display, where these facilities include character storage, refresh timers, and character generation. If the display is unchanging, then no information need be exchanged between the display unit (which includes the controller) and the host computer.

All changes to a display are the result only of a transfer of information from the host computer to the display unit. Some changes may be the result of user action at the console, while all the other changes are due to changes in the aircraft state. As an example of the latter, the displayed value of the aircraft's present latitude will vary as the aircraft moves.

When a display change is made in response to a user's action, that change should not be delayed in excess of 0.2 second.

b. Methods of Analysis

Of concern are (1) the amount of CPU core storage required by the set of four displays, and (2) the amount of CPU time required to service that set of displays.

In considering the core storage requirement, we see the need only to store (1) the format statements associated with the set of 60 pages, and (2) the programs that are needed to create the display list. Excluded from this requirement are (1) storage for display, thence not part of the CPU, and (2) storage of variables, for they are assumed to be associated with their own programs, which are separate and distinct from any display activity.

In addition, all code, tables, data, etc., are assumed to be continuously resident in the core, with no overlaying taking place through the use of a mass storage device, such as a disk or tape unit. All code is re-entrant, so that only one copy is kept of each display program (assuming no redundancy for purposes of reliability); thus only one copy is kept of each type of page format. Only one buffer area for display list assembly is provided to serve the entire set of displays. However, a small buffer is provided for each display to hold input from the consoles.

For CPU utilization, the matter is simpler. Demands for CPU time are seen as coming from only two sources-creation of a new display list, and servicing of console inputs. Excluded as a demander of CPU time is refresh control, for we have allocated that task solely to the display controller. As a first cut, we will assume that any change in the display list requires a complete recomputation to the display list, thus providing an upper bound on the CPU utilization for that task.

On the other hand, we will consider the servicing of console inputs to be small enough, with respect to the display list creation task, to be ignored. We reason that as a worst case the display list will be changed ten times per second, whereas the console input will not exceed one keystroke per second. If the CPU time to process a keystroke does not exceed the time to create a display list, we can ignore the console input load.

The estimation that we develop here is for the CPU time to construct a textual display list that accounts for all 16 character positions for each of the 16 lines. There will be an average of nine variables per page, with an average of five characters per variable.

Multiplying that derived time by the rate at which new lists are constructed provides us with the continuous computing load. We are anticipating that the rate will be between three to ten new displays per second.

A flow chart of the process is given in Figure 3. It is the outer loop, boxes 1 to 11, that is traversed once each time a new display is created. Box 11 determines the frequency of display creation; in particular, functions of the box include initiating a new display if one of the variables currently being displayed (e.g., present latitude) changes, or new input is received from a console. As it will develop, box 11 serves all four displays, but permits only one display to be created at one time. Because of this, only one 256-byte buffer area need be in use at any time, in which to build a display list.

The inner loop, boxes 2 to 10, is traversed an average of 19.5 times for each display creation. This follows from an analysis of the Collins type of display, in which an average of 19.5 items appear on the screen or control it. These items include words and phrases such as "VIA," "LAT/LONG," and "INSERT," and variable information such as "47N26.8," "312^0," and "LGA." Each type of page to be displayed can be described by its own unique list of items. We assume here that each entry of the list contains: (1) one eight-byte address to specify where on the screen the item is to be displayed, and (2) a one-byte pointer to the list of all possible displayable quantities.

Based on the Collins SST discussions, we assume that there will be a maximum of 60 types of pages; hence 60-page lists will be concurrently held in core. We also assume that the list of displayable quantities is 170 in number. This approach to organizing the page formats has been chosed because of its efficient use of core storage. It seems particularly applicable to this environment in which the displayed material is highly constrained.

The execution of box 1 causes the program to clear all 256 bytes of the display list area to blanks. (Subsequently, box 8 causes certain of these bytes to be replaced by other characters.) After thus clearing the area the program goes to the top of the selected page list, proceeding down item by item until the end of the list is encountered.

Following the pointer of the current item, via box 2, the program enters the list of displayable quantities and either (boxes 3 to 5) determines that the quantity is text or (boxes 3 to 7) determines that the quantity is a datum, in which case a pointer is again followed.

c. Estimates

1) Memory Requirement

Data for SIDs, STARs, MAPs	7,800 32-bit words	
Page Lists	2,340 bytes	860 32-bit data words
Item descriptions	836 bytes	
Display create area	256 bytes	
Program instructions	640 bytes	

2) CPU Utilization

Number of short instructions executed per display page = 1,500

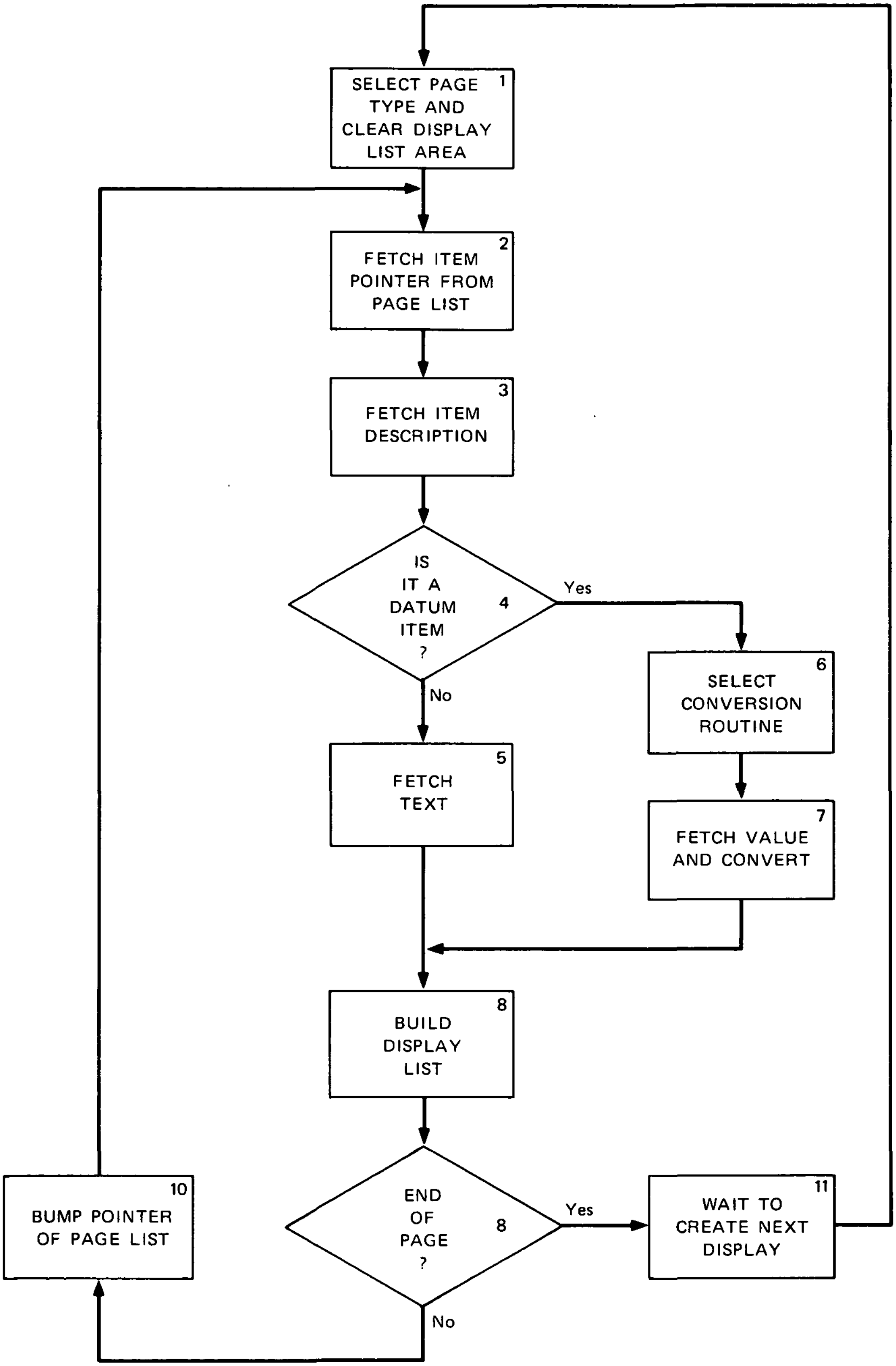
Number of long instructions executed per display page = 100

At 4.0 microsecond per short instruction and 16.0 microseconds per long instructions:

(1500) (4.0) + (100) (16.0) = 7600 microseconds to create one new page

(10) (7600) = 76,000 microseconds per second, or 76 milliseconds per second to create 10 new pages.

This is equivalent to a 7.6 percent utilization of the reference computer.


SELECT PAGE TYPE AND CLEAR DISPLAY LIST AREA
1
FETCH ITEM POINTER FROM PAGE LIST
2
FETCH ITEM DESCRIPTION
3
IS IT A DATUM ITEM ?
4
Yes
No
FETCH TEXT
5
SELECT CONVERSION ROUTINE
6
FETCH VALUE AND CONVERT
7
BUILD DISPLAY LIST
8
END OF PAGE ?
8
Yes
No
BUMP POINTER OF PAGE LIST
10
WAIT TO CREATE NEXT DISPLAY
11

SA-1406-28

FIGURE 3 FLOWCHART FOR UPDATING DISPLAYS OF TEXT

G. Collision Avoidance Systems (CAS)

1. General Considerations

Collision Avoidance Systems (CAS) have for the past few years been the subjects of much debate and analysis, and of a limited amount of actual test. To date, no definitive decision has been made, and this estimation of an on-board data-processing facility must therefore be based on many assumptions and judgment factors. Three of the contending CAS systems are:

- The "Time/Frequency" (T/F) System (Airborne)
- The "Separation Control of Aircraft By Nonsynchronous Techniques" (SECANT) System
- A Ground-based Time-frequency System.

The airborne T/F system is the one that is used primarily in this estimation of airborne data-processing requirements because it is both politically and technologically further developed than the others. For SECANT, only simulations and very-limited flight-test operations are reported to date, and for the ground-based T/F concept, the major deterrent appears to be the basic operational preference for basic CAS independence from ground-based equipment. For background purposes, the three systems are summarized briefly below.

a. The Time/Frequency (T/F) System

The Air Transport Association (ATA) has tested and recommended implementation of the T/F system. This system makes each of 2000 time "slots" of 1.5 milliseconds exclusively available for the transmissions of one of as many as 2000 equipped (cooperative) aircraft in an "area" determined by aircraft altitudes, radiated power, etc.* The basic "epoch" (cyclic period) of the system is 3 seconds. Each aircraft's transmission is timed by an airborne clock accurate to approximately 0.2 microsecond. Reception of that transmission by each of the other "cooperative"

* Actually, some of the 2000 slots would be used for control purposes, for terrain-avoidance installations, etc.

aircraft enables the receiving aircraft to determine several parameters relative to the transmitting aircraft (among others):

- Altitude (A)--via analog encoding.
- Range (R)-- via (absolute) time of reception of the initial pulse train.
- Range Rate ($\dot{R}$)--via Doppler shift.

Potential collision hazards are flagged via on-board computations based primarily on these three parameters, a prime factor designated as Tau ($\tau = R/\dot{R}$) indicating "time to minimum-distance" passage.* If the two aircraft are at or near equal altitudes, a small value of Tau (e.g., a few seconds) indicates a potential hazard, and evasive UP/DOWN maneuvers are commanded.†

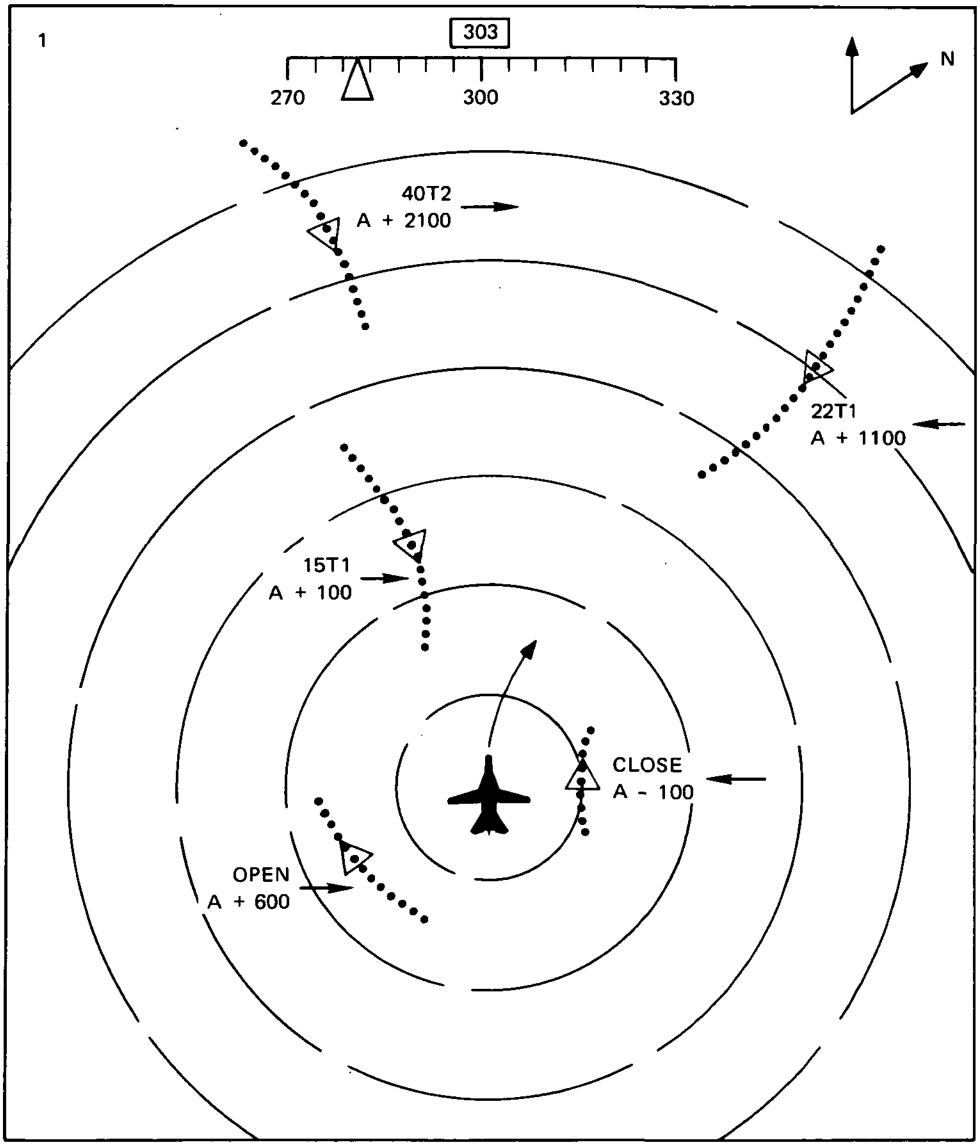
There is still much debate as to the real potential effectiveness of the T/F system--particularly for operations in terminal areas as contrasted to enroute operations. Incorporation of an azimuth-defining capability (antenna) to enable LEFT/RIGHT as well as UP/DOWN evasive maneuvers is being proposed, but this constitutes yet one more indeterminant at this time.

In the system as now conceived, an aircraft's T/F unit is not required to store any historical data on any of the other aircraft from epoch to epoch (3-sec intervals)--all decisions are based upon "current" data. However, if an azimuth capability were to be incorporated, some measure of history would necessarily be incorporated into the system. It appears that the azimuth capability coupled with a graphics CRT display would greatly enhance the CAS effectiveness of the T/F system (and others)--particularly for terminal-area operations. A sketch of a conceptual CAS display is presented in Figure 4.‡ A parallel ILS approach situation is

*In practice, Tau "zones" would be used instead of the simple Tau parameter

†For those cases where $\dot{R} < 80$ knots, a range/altitude evaluation would be used.

‡The Graphic Display Unit (GDU) potentially uses the same general type of display. It is possible that in practice, the same display hardware could be used for simultaneous presentions of the GDU and the CAS data.



SA-1406-29

FIGURE 4 A CONCEPTUAL COLLISION AVOIDANCE SYSTEM (CAS) DISPLAY

illustrated (the host aircraft and the one directly to the right), this type of situation being accommodated readily with pilot interpretation of the pictorial display, whereas the (nearly) equal altitudes of, and the short range between the two aircraft would be relatively difficult to accommodate reliably with computer logic only.

b. The SECANT System

This system is based on interrogation/response sequences initiated on a completely asynchronous basis by each cooperative aircraft, responses being received from all other aircraft within range. Interrogation "probes" are all of 1-microsecond duration, the average repetition rate being 1000/second (the actual rate is randomly modulated to monimize interference among transmissions of all cooperative aircraft).

Assessments of potential collision hazards are to be based on azimuth and altitude information, on range (R), and on range rate ($\dot{R}$). In the SECANT situation where many aircraft are transmitting and receiving on common channels, statistical integration of received signals is required--for this purpose, "range cells" or "data bins" of approximately 500 feet (1μs) are established throughout the range of interest, counts being accumulated for 100 ms (100 probes and response periods). Analyses of R and $\dot{R}$ are based on the accumulated counts in these bins, consistent "hits" approaching bin counts of 100, but "fruit" (responses to other interrogations) hopefully limited to counts of 5 or 10 maximum. As is the case of the T/F system, SECANT threat analyses are proposed to be based in part on Tau ($\tau = R/\dot{R}$), the "time to minimum-distance" passage. In addition, the more sophisticated SECANT systems proposed will incorporate the following features:

- Air/ground data links to alert Air Traffic Control (ATC) personnel of potential hazards, and air/air data links for coordination between two aircraft.
- A CRT display of potential threat aircraft to aid the pilot in planning optimum evasive maneuvers (in only the most sophisticated form, the Traffic Monitoring System (TMS)).

Because of the early stage of development of the SECANT system, only very rough estimates are made in this report of the EDP requirements. In any event, however, the type of requirement should be noted because of the demand for more processing time, and for additional core storage, as considered briefly in Section IVG3 below.

c. Ground-based Time/Frequency Systems

In 1969, Autonetics (North American Rockwell)submitted its final report to the Federal Avaiation Administration on "Analysis of an Advanced Time-Frequency National Air Space System Concept." Separation assurance (or collision avoidance) was included within that concept, the basic technique proposed being as follows:

- Aircraft would determine position and velocity on a passive basis, using receptions from perhaps three or four ground stations.* These data and altitude would be sent to ground-based processing stations via data link.
- The ground-based processing stations would analyze these data from all aircraft with the goal of flagging all potential collision hazards, and transmitting to the involved aircraft the commands for proper evasive maneuvers.

It appears that this system concept has not been generally accepted by the aviation community, one of the major factors being the desire for an airborne CAS capability basically independent of ground-based facilities--except as a backup mode.†

*Use of the passive mode of operation would preclude the possibility of system saturation in the sense that the number of slots required (for the ground stations) would be independent of the number of airborne aircraft. In fact, the epoch could undoubtedly be shortened considerably.

†However, a ground-based system of CAS-like service is proposed in connection with the Discrete Address Beacon System (DABS). As discussed in Sec. IV-H on digital data communications, it has been suggested that Intermittent Positive Control (IPC) vectors based on radar-derived data be transmitted to aircraft in connection with DABS operations to eliminate potential collision situations.

In the event that such a ground-based T/F system were to be implemented, the airborne-computer requirements for processing of the T/F data would be considerably less than those of the independent airborne T/F system. However, for overseas operations, the system would necessarily revert to the air-to-air mode (unless enough satellites were available to accommodate the aircraft-passive mode of operation). The ground- or satellite-based modes of operation would require increased data link traffic, but this load on the airborne computers would not be significant relative to other requirements.

2. Summary of Analysis of the Requirements of the T/F Collision Avoidance Systems (CAS)[*]

Detail on the analysis of the data-processing requirements of the Time/Frequency type of CAS is provided as one of the illustrative examples in Appendix B.

a. Core Requirements (Not Including Subroutines)

● Program (Instructions)	450
● Buffer/Data[†] (Words at 32 bits each)	600
● Symbols (Common to Graphic Displays)	--
● Supplementary (Program)	100

b. Number of I/O Words per Second (Packed at 2 Parameters/Word)

● Inputs	1400
● Outputs	225

c. Processing Operations per Second

	No./Second	Comp. Time
● Shorts	13,892	5.6%
● Longs	1,570	2.5
● Supplemental (Shorts)	1,000	0.4
		8.5%

[*] As discussed in Section IV-G-3 below, the data-processing requirements would be higher if the SECANT system were to be implemented.

[†] Range/azimuth (R/θ) or X/Y data for each point packed into one work. Provision is made for accommodating as many as 12 potentially hazardous aircraft simultaneously.

3. Possible Data-Processing Requirements of the SECANT Collision Avoidance System (CAS)

If developments were to indicate that there may be a high probability of implementation of the SECANT system, it would be essential that a more detailed analysis of that system be made relative to its implications on total data-processing requirements. There is a possibility that the SECANT system could require substantially more processing time than the T/F system because of its basic "bin" mode of operation:

- A significant fraction of approximately 600 such bins might be incremented or decremented every millisecond (by either valid returns or by "fruit" pulses).
- The accumulated counts of all 600 such bins would necessarily be inspected every 100 ms, and some of them subjected to detailed processing pertaining to range, range rate, etc.

Without making a detailed system analysis (including statistical factors), it is not possible to specify a reliable estimate of SECANT requirements. However, it is believed that they might amount to as much as four or five times that of the T/F system, this corresponding to approximately 40 percent for the reference computer assumed (at this early point in the study).

In regard to hardwire, "bins" would be required for each of the 500-foot (1 μs) range increments along a given azimuth. Assuming that count data are to be retained for only one azimuth at a time, and further assuming a maximum range of 60 miles, 600 (up-down) counters would be required for the pulse integrations. If implemented by arithmetic addition/subtraction rather than by special hardware, this would correspond to 150 words of memory (packed at four counters per word). Also, a more complex SECANT program might require an additional 500 program instructions.

H. Data Communication Systems

1. General Considerations

Digital data communications of future jet transports will involve two major categories:

(1) Internal (Aircraft Inter-unit)

- The acquisition of the magnitudes of various parameters as sensed by a variety of transducers.
- The transmission of computer-processed parameters for aircraft control, for display, and for storage.*

(2) Air-ground-air

- The reception from, and transmission to, ground-based Air Traffic Control (ATC) stations of messages concerned with flight plans and clearances, communication/navigation tuning, position reporting, weather and terminal-area conditions, etc.
- The reception from, and transmission to, the airline ground stations of messages concerned with general flight information, aircraft performance, passenger service, etc.

Estimates of memory and computer processing requirements of both categories of digital data communications are summarized below in Section 2, and are tabulated in detail in Appendix B.† However, it is to be noted

* Storage of data refers primarily to the functions of the Aircraft Integrated Data System (AIDS).

† As will be the case with a number of the other data-processing operations, the digital data communication function will most probably experience peak activity during the approach/landing phase of flight--the time estimates of this section are therefore derived for that phase of flight.

that these estimates do not include provision for:

- Tests for accurate transmission of the data (e.g., parity checks, computation of validity of longitudinal parity checks for transmission of blocks of data).
- Cross-channel data consistency checks, majority voting or weighting, etc.
- Data reasonability tests against immediately preceding values (of sampled parameters) and/or against prestored limits.
- Timing functions to control scheduling of sampling input data, and transmitting output data.

It is assumed that all such tests and procedures are to be performed on a centralized basis consistent with overall system philosophy and architecture.*

The "Internal" digital data communications between the computer and the various on-board sensors, actuators, displays, and "preprocessing" units are assumed to be based on reasonably conventional, buffered block-transfer techniques. It is assumed that all analog-to-digital and digital-to-analog conversions are performed as required by built-in hardware at individual units or preprocessors, or by conversion hardware in an I/O preprocessor shared by a number of channels.

Equipment and procedural specifications for air-ground-air digital data communications are in an embryonic state at the present time. However, it is generally acknowledged that such a facility is required--

* These types of processing may constitute significant requirements. For example, Collins estimates a 20 percent computer-time requirement for "I/O Control"--a figure significantly higher than seems appropriate for simple transmission of the data involved in its AINS-70 RNAV System.

not because the absolute magnitude of the data to be transmitted is great, but because voice transmission of even small blocks of data between a ground station and a number of aircraft sharing a "party-line" channel can become tedious and even confusing. This is particularly true in terminal area operations where a significant number of aircraft must be (sometimes) stacked in holding patterns and vectored into "windows" of the approach paths for one or more runways of one or more major air terminals. However, the same general type of problem exists for enroute aircraft under control of a "Center."

There are two primary contenders for air-ground-air digital data communication:

- Air-ground-air Data Link System[16]
- Discrete Address Beacon System (DABS)[17].

The data link system accommodates transmission of data blocks of as many as 68 characters of eight bits each accompanied by approximately 40 "prekey," control, and check characters.

The DABS system of data transmission is a component of an interrogation/transponder system whereby the ground facility would as appropriate "address" a specific aircraft to send approach-vector information, to send Intermittent Positive Control (IPC) vectors to eliminate potential collision situations, to request position/altitude data, to control enroute operations, etc.*

* The DABS system would eventually replace the current ATC Radar Beacon System (ATCRBS) wherein each airborne transponder responds to all received interrogations, providing a coded (1 of 4096) "category" self-identification and possibly altitude information. However, an attempt will be made to achieve some measure of compatibility between the two systems so that with reasonable expense of modification, an ATCRBS transponder will be able to limit its responses to only those interrogations directed to its own discrete address, thereby precluding high costs of complete unit replacement.

Each transponder responds to only those interrogations discretely addressed to it. The system thereby minimizes possibility of: (1) garbling between responses from two transponders at similar ranges; (2) possible system saturation in congested areas.

Message formats have not yet been specified, but unique encoding of ATC-related commands will facilitate minimizing message length to perhaps as few as 100 to 200 bits including address--for purposes of analysis, six words of 32 bits each are assumed here. The air-ground messages will undoubtedly be shorter. The message rates are estimated to be of the order of "...once or twice every few seconds by (each of) one or two interrogator sites."* For purposes of analysis, a peak (terminal-area) "burst" rate of four messages/second is assumed--no aircraft crew could act upon or even monitor more information than would correspond to these assumptions. However, it is to be noted that the airborne DABS system must receive and decode the addresses of all interrogations to determine whether or not a response is required.† A peak interrogation rate of 800/second is assumed.

The estimates of time requirements are based on these assumptions as the worst-case, terminal area conditions. Some projections of requirements as high as a 10-kHz data rate have been made, but these seem to be excessive--even with some measure of automatic control included (full flight path control from the ground certainly will not be implemented within the next ten years).

* See Ref. 17, p. II-15.

† It is assumed in the analysis of this section that the DABS hardware provides the facility for address decoding, thereby relieving the transport computer of this function. Because of the immediate-response requirement for some modes of DABS functions, an unreasonable interrupt or polling requirement would otherwise be imposed upon the computer system.

For enroute operations, ATC and weather-related messages will be relatively long, but the requirement for immediate transmission and response much lower. The average data transmission and processing rates must therefore be negligible, and in general, the priorities, low. The same philosophy applies also to company-directed messages such as those related to aircraft performance and maintenance requirements, and to passenger-service (air, auto, hotel reservations, connecting flight delay request, etc.).

2. Summary of the Analysis of the Requirements for Digital Data Communication

Detail on the analysis of the data communication requirements is provided as one of the illustrative examples in Appendix B. In summary, those requirements are as follows.

a. Memory Requirements (Not Including Subroutines)

	Internal	Air-Ground-Air
● Program (Instructions)	210	450
● Data and Buffers (Words at 32 bits each)	400	112
● Supplementary (Program)	70	100
● Common Buffer (Words at 32 bits each)	50	

b. Number of Operations per Second

● Inputs	6,450
● Outputs	3,700

c. Instructions per Second and Time Requirements*

	Internal		Air-Ground-Air	
	No./Second	T_{comp}	No./Second	T_{comp}
● Shorts	5,660	2.3%	500	0.2%
● Supplemental (shorts)	1,340	0.5%	120	0.05%
Subtotals	7,000	2.8%	620	0.25%

Grand Totals: 7,620 Instructions/second
3.1% of computer time.

*These estimates are made for the approach/landing phase of flight (worst-case conditions). The figures for the air-ground-air requirements would be somewhat higher if the decision were to be made to implement processing and graphic display of ground-derived information such as the locations and severity of weather and turbulence information, and the positions (and vectors) of aircraft presenting potential collision or congestion threats (not already derived via CAS). However, the absolute increase in the total load would not be significant--even if the percentage increase in the air-ground-air requirements were to be as high as 30 or 40 percent. The requirements for processing the added internal data communications would also be increased by a small amount.

I. The Aircraft Integrated Data System (AIDS)

1. General Considerations

There is a well-defined trend toward monitoring, recording, and even in-flight analysis of an increasing number of parameters in commercial aircraft. The federal Aviation Agency (FAA) and Aeronautical Radio Inc. (ARINC) are properly concerning themselves with only the basic AIDS system specifications on critical parameters, connector configurations, and the like, leaving the specification of the majority of parameters and modes of utilization to the airlines themselves.[18]

A substantial number of systems have already been installed in operational aircraft, and the modes of implementation vary widely. For example, BOAC contracted with Plessey for implementation of a system that is strictly a recording system. A cassette is provided for 22-track recording on a 1-inch tape for a total recording period of 40 hours. BOAC analyzes these data at ground installations and uses the results primarily as positive or negative feedback to the crew in terms of their performance (this reportedly reduces the number of actual flight hours required for crew proficiency tests). In the case of Hamilton Standard systems developed for KLM, and the Teledyne units for TWA and others, emphasis is on aircraft performance and maintenance, and some provision is made for on-board access to the data. Though apparently not yet implemented, plans are in the works for programming the computer to perform analyses based on excessive deviations of critical parameters and/or performance figures based on processing on a multiple-parameter basis. In the event that a potential problem area is flagged by these on-board analyses, the flight engineer could either immediately take corrective action or radio ahead to arrange for prompt service at the destination facility as appropriate.

The various philosophies of vendors and airlines also imply different data-recording procedures. In the case of the Hamilton Standard equipment, recording is presumably continuous. In the case of Teledyne, the AIDS data are recorded continuously on a 5-minute loop, new data being written over old data after each such period. In this manner, a complete history of the past 5 minutes of aircraft operation is available, these data being recorded for retention only at specific times throughout a trip. In the case of the AIDS system developed for TWA, a complete sampling of the outputs of approximately 360 sensors is automatically recorded on a second unit at each of several scheduled times during engine startup, during takeoff and climb, during cruise, during descent and landing, and during engine shutdown. In addition, the flight engineer can specify recording of these parameters via his Cockpit Control/Display Unit at any time and, in the event of detected problems, can transfer the previous 5 minutes of data from the loop recorder to the main recorder for later analysis.

In such an environment of developing technology, it is possible only to use current information as a narrow frame of reference for what are hopefully reasonable extrapolations for the following 10-year period. The estimates here are based primarily on information on current Teledyne and Boeing systems. (Additional information pertaining to the AIDS system installed by Northrop aboard the Air Force C-5A transport should be available soon.) Unfortunately, the data on hand do not provide a basis for direct scaling of parameter storage and iteration-rate requirements, but it is believed that the following represents a reasonable estimation.

a. Requirements for Data Acquisition and Storage

For the Teledyne loop recorder concept, the requirements for data acquisition (and storage) would be approximately as follows:

Data Type	No.	Definition (Bits)	Iteration Rate (No./ Second)	Average Bits/ Second
Engine Parameters	160	12	1/4	480
Flight/Aircraft Instruments	100	12	1	1200
Acceleration	6	12	4	288
Navigation/Communication	50	12	1	600
Discretes	100	1	1	100
Totals	400^+	--	--	2700^-

For purposes of establishing computer requirements, the above can be more properly interpreted on a word rather than a parameter basis. It is assumed that full parameters and discretes will be packed into words of 32 bits each (relative to computer processing). This would correspond to blocks of approximately 100 words accessed and 100 words stored per second on the average.* Data acquisition, packing, and storage programs are relatively simple, so it is assumed that program requirements for this phase of AIDS will be of the order of only 200 words.

b. Requirements for On-board AIDS Analysis and Processing

There are at least three basic classes of techniques of on-board AIDS analyses that would place demands on the computer system over and above those related to the simple recording function:

- Interrogation of specific parameter values by the flight engineer
- Limit checks
- Multiparameter analyses.

To date, experience in these areas has been limited or perhaps nonexistent. However, it is virtually certain that such techniques will be used for the more critical parameters in the time frame of interest. Estimates are as follows for implementation of these techniques:

(1) Interrogation of Specific Parameter Values by the Flight Engineer--Program and computer time requirements for this type of facility are minimal. However, for any such system to be meaningful, the displayed data must be in terms of actual engineering quantities. This requires storage of a scaling factor for each parameter to be subject to interrogation and display. If it is

*The 2:1 ratio of storage versus acquisition rates is based on the assumption that during the "worst-case" data-processing requirements, recording on both the loop and the permanent storage tapes will be programmed.

assumed that 300 parameters must be accommodated, and that the scaling factors can be packed two to a word, the total storage requirement would be approximately 150 words, the program being about 50 words in length. Computer time and priority requirements would be minimal.

(2) Limit Checks--It would appear that simple max/min limit checks will be of considerable value to the flight engineer. Based on the assumption that both the maximum and minimum values (appropriately scaled) can be packed into one word for each parameter, 300 words of storage would be required. In addition, it assumed that 50 words would be adequate for the program. There should be no need (peak) for a limit check cycle oftener than perhaps once per 4 seconds. A low priority will be adequate.

(3) Multiparameter Analyses--It is assumed that the number of parameters subjected to in-flight analyses will be relatively small, so that a memory requirement of the order of 300 instructions and 200 data words will be appropriate. It is further assumed that one multiparameter computation will be made only once per 5 seconds during the critical activity period.

2. Summary of Analysis (AIDS)

a. Memory Requirements (Not Including Subroutines)

● Program (Instructions)	550
● Data and Buffers (Words at 32 bits each)	650
● Supplementary (Program)	100

b. Number of I/O Words per Second

• Inputs	100
• Outputs	200

c. Processing Operations per Second

	No./Second	Computer Time*
• Shorts	1,000	0.4%
• Longs	60	0.1%
• Supplementary (Shorts)	200	0.1%
Total		0.6%

J. Instrument Monitoring, "Other" Systems Management and Monitoring, and Life Support System Monitoring and Management

The Instrument Monitoring function is not currently mechanized to any degree in today's civil aviation fleet. With the exception of power- or signal-interruption warnings, the monitoring task is currently done by the aircrew. The same is true of "other" and "Life Support" system management functions, e.g., fuel, electrical power, hydraulic, cabin temperature and pressure, and passenger service functions. However, aircrew workload considerations point to a need for mechanizing those functions that need to be done on a frequent or continuous basis. SRI has recently completed a feasibility study of a system concept for identifying malfunctions in flight control data instrumentation.[2] A digital simulation was conducted, and

*These estimates of computer time requirements are based on the assumption that no addressing or control is required of the computer during the data acquisition phase--all scanning operations being automatically provided by the Flight Data Acquisition Units (FDAUs). This is in contrast to some of the Teledyne system concepts where the computer does provide the addressing function, where amplifier gain must be a slewed for specific data accesses, etc. It is believed that such procedures involve unnecessary complications and demands on computer time, and are therefore not used as a basis for these estimations of requirements on the computer system.

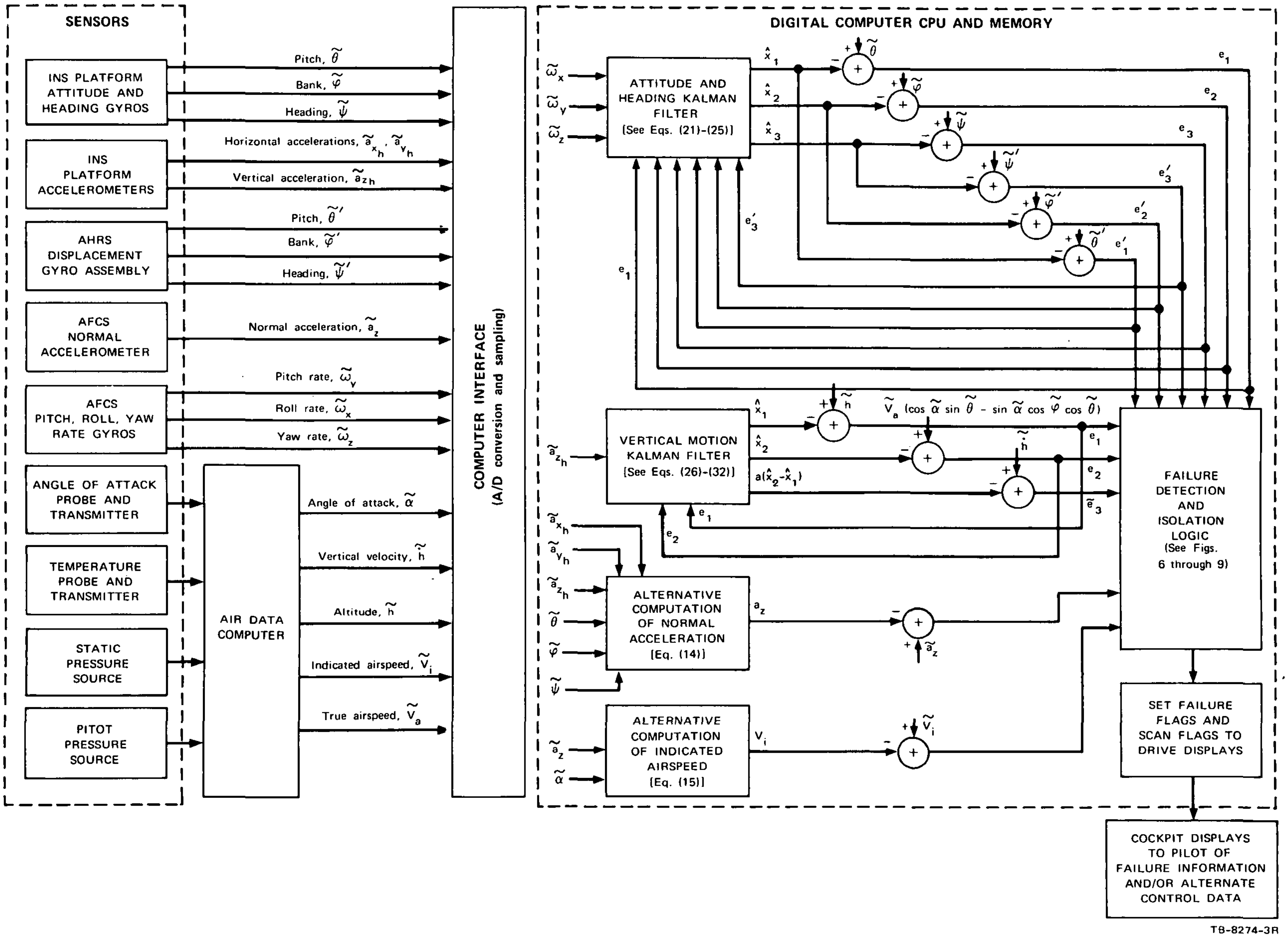

SENSORS
INS PLATFORM ATTITUDE AND HEADING GYROS
INS PLATFORM ACCELEROMETERS
AHRS DISPLACEMENT GYRO ASSEMBLY
AFCS NORMAL ACCELEROMETER
AFCS PITCH, ROLL, YAW RATE GYROS
ANGLE OF ATTACK PROBE AND TRANSMITTER
TEMPERATURE PROBE AND TRANSMITTER
STATIC PRESSURE SOURCE
PITOT PRESSURE SOURCE
Pitch, $\tilde{\theta}$
Bank, $\tilde{\varphi}$
Heading, $\tilde{\psi}$
Horizontal accelerations, $\tilde{a}_{x_h}$, $\tilde{a}_{y_h}$
Vertical acceleration, $\tilde{a}_{z_h}$
Pitch, $\tilde{\theta}'$
Bank, $\tilde{\varphi}'$
Heading, $\tilde{\psi}'$
Normal acceleration, $\tilde{a}_z$
Pitch rate, $\tilde{\omega}_y$
Roll rate, $\tilde{\omega}_x$
Yaw rate, $\tilde{\omega}_z$
AIR DATA COMPUTER
Angle of attack, $\tilde{\alpha}$
Vertical velocity, $\tilde{\dot{h}}$
Altitude, $\tilde{h}$
Indicated airspeed, $\tilde{V}_i$
True airspeed, $\tilde{V}_a$
COMPUTER INTERFACE
(A/D conversion and sampling)
DIGITAL COMPUTER CPU AND MEMORY
$\tilde{\omega}_x$
$\tilde{\omega}_y$
$\tilde{\omega}_z$
ATTITUDE AND HEADING KALMAN FILTER [See Eqs. (21)-(25)]
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\tilde{\theta}$
$\tilde{\varphi}$
$\tilde{\psi}$
$\tilde{\psi}'$
$\tilde{\varphi}'$
$\tilde{\theta}'$
e_1
e_2
e_3
e_3'
e_2'
e_1'
e_3'
e_1
+
−
$\tilde{a}_{z_h}$
VERTICAL MOTION KALMAN FILTER [See Eqs. (26)-(32)]
$\hat{x}_1$
$\hat{x}_2$
$a(\hat{x}_2-\hat{x}_1)$
$\tilde{h}$
$\tilde{V}_a$ (cos $\tilde{\alpha}$ sin $\tilde{\theta}$ − sin $\tilde{\alpha}$ cos $\tilde{\varphi}$ cos $\tilde{\theta}$)
$\tilde{\dot{h}}$
e_1
e_2
$\bar{e}_3$
e_1
e_2
$\tilde{a}_{x_h}$
$\tilde{a}_{y_h}$
$\tilde{a}_{z_h}$
$\tilde{\theta}$
$\tilde{\varphi}$
$\tilde{\psi}$
ALTERNATIVE COMPUTATION OF NORMAL ACCELERATION [Eq. (14)]
a_z
$\tilde{a}_z$
$\tilde{a}_z$
$\tilde{\alpha}$
ALTERNATIVE COMPUTATION OF INDICATED AIRSPEED [Eq. (15)]
V_i
$\tilde{V}_i$
FAILURE DETECTION AND ISOLATION LOGIC (See Figs. 6 through 9)
SET FAILURE FLAGS AND SCAN FLAGS TO DRIVE DISPLAYS
COCKPIT DISPLAYS TO PILOT OF FAILURE INFORMATION AND/OR ALTERNATE CONTROL DATA
TB-8274-3R

FIGURE 5 FLOW DIAGRAM OF SIGNAL PROCESSING FOR INSTRUMENTATION MONITORING TECHNIQUE

the computational requirements found in that effort are used here for the Instrument Monitoring Function, and scaled down in size and frequency for the "other" and "Life Support" System Monitoring and Management Functions. The estimates appear in Table 3, Section III. The flow diagram of Figure 5 illustrates the logic and type of computation of the Instrument Monitoring algorithm assumed.

K. Engine Systems Control and Operation

1. Introduction

The gas turbine engine basically consists of a set of combustion chambers, one or more turbines, one or more compressors, an inlet duct, and an outlet duct. At a controlled rate, fuel is continuously sprayed into the combustion chambers, where the fuel burns in the large volumes of air entering the chamber from the compressors. The resulting hot gases flow through the turbines and thence discharge to the atmosphere via the outlet duct. The turbines, mechanically coupled to the compressors, provide the power to compress the air entering the inlet duct and prior to delivery of the air to the chambers.

It is interesting that only about 1/4, by weight, of the air entering the inlet duct is used for combustion; the remainder is used to cool the combustion chamber surfaces and to cool the burned gases before they enter the turbines. Also, only about 1/4 of the power generated inside a jet engine is available to produce thrust to propel the aircraft; the remainder is used to drive the compressors.[19]

The pilot's primary concern is that the engines develop the amount of thrust he desires. That desired amount, in terms of percent of maximum thrust that the engine will consistently deliver, is set into the fuel control system, by his positioning the fuel control lever. By means of hydraulic servos, gears, cams, levers, and valves, the hydro-mechanical fuel control systems regulate the flow of fuel to the engine. These systems attempt to satisfy the thrust demand indicated by the fuel control lever. For a variety of reasons, the resultant thrust may differ from the demand thrust.

The control system senses such variables as compressor inlet temperature compressor discharge pressure, compressor rotational speed, and combustion chamber pressure. These variables are used to control the rate of change of fuel flow and the rate of fuel flow, for considerations of engine life and of operational safety and efficiency may dictate that the engine thrust differ from that commanded by the control lever setting. For these reasons, the control system will also constrain the rate at which the engine may accelerate or decelerate. In spite of the fuel control system, thc engine can still get into unsatisfactory conditions, such as excessive temperature at the turbine inlet, improper fuel-air mixture ratio, and compressor stall. It is the pilot's responsibility to monitor the engine instruments to avoid these bad conditions, and to afford recovery when they occur.

2. Some Observations Regarding the Present Forms of Fuel Control Systems

We can gather some impressions of present-day fuel control systems by an examination of the mechanism that is used to control the JT8D commercial turbofan engine; there is one such controller for each engine. That controller receives six input signals and has 16 adjustments. The inputs are (1) fuel control lever setting, (2) fuel shut-off control lever setting, (3) compressor inlet temperature, (4) compressor discharge pressure, (5) compressor rpm, and (6) fuel temperature. The adjustments provide for trimming of the input signals, limiting minimum idle speed, limiting minimum fuel flow, and adjusting the acceleration and deceleration profile.

The "heart" of the hydromechanical controller is a cylinder with a surface shape that is quite complex, in that the radial distance to the surface is a function of the surface's angular position around the cylinder's axis Changes in compressor inlet temperature cause the cylinder to rotate about its axis, while changes in compressor speed cause it to translate along the line of its axis. Cam followers press against the cylinder's surface and affect the fuel flow. The cylinder's surface represents, in some way, the acceleration profiles to which the engine is constrained. These profiles are chosen to minimize the instances of turbine over-temperature and of compressor stall.

Direct replacement of these mechanical controllers by a digital computer appears to provide only limited advantages, e.g., possibly lower fabrication costs (especially considering the complex cams) and increased flexibility in that several sets of acceleration profiles might be available to meet special operational requirements.

In current practice, there are several instruments in the cockpit that indicate several engine parameters not monitored by the fuel control system. These include:

- The speed of the low-pressure compressor for dual compressor engines
- Exhaust gas temperature (EGT)
- Engine pressure ratio (EPR).

These parameters are quite useful in managing engine thrust; other indicated parameters, such as those pertaining to the fuel and oil systems are primarily used to detect actual or incipient malfunctions, and are further discussed here.

EGT is used as an indirect measure of turbine inlet temperature. Excessive inlet temperature can result in markedly decreased engine life, if not major engine harm. The extreme heat at the turbine inlet prevents the use of temperature sensors at that point, thus forcing the use of EGT as an alternate form of measurement.

The EPR is a measure of engine thrust; it is the ratio of total turbine discharge pressure to the total pressure of the air entering the compressor.

By movement of the fuel control lever, in conjunction with the measured EPR, the pilot can manage the thrust production of each engine. In the process he may also adversely affect the engine, either through overheating, compressor stall, or other misadventures. It is the pilot's responsibility to monitor EPR, EGT, and compressor speeds to ensure proper engine operation.

In the case of engines operating at sonic speeds, control of the inlet and outlet ducts also comes into play. It appears that under normal operations the functions are under automatic control. However, it is anticipated that the pilot will be responsible for control of the duct geometrics when serious changes in operating conditions occur, such as engine flameout, compressor stall, or inlet unstart. Indications are that (1) insufficient instrumentation is currently available to permit the pilot to analyze the situation in order to generate the proper response, and (2) given that instrumentation, the pilot's response may be too slow. Slowness of response, it is believed, would primarily introduce serious diseconomies in the operation of the aircraft, but not necessarily result in a dangerous condition.

3. Experimental Digital Computer Replacement of the Hydromechanical Controller

a. Observed Results

At the NASA Lewis Research Center, a J-85-13 turbojet engine on a sea-level test stand was placed under the control of an SEL 810B computer.[21,22] The computer was programmed to perform the same functions, no more, no less, than the equivalent hydromechanical controller. In Ref. 20, it is stated that "[The computer] was shown to match the hydromechanical response at control update intervals to 30 milliseconds. At extended intervals, some deterioration in response was noted, but stable operation was demonstrated to an update interval of 200 milliseconds."

Their computer was equipped with 16K words of core storage, with 16 bits plus parity for word size, and a 750-μs cycle time. Add or subtract time is 1.5 μs, while multiply time is 4.5 μs.

Analog input channels were sampled 20K times per second, with 12 bits plus sign out of the digitizer for each sample. It would appear that only a few of these samples are actually used by the program, for the latter could not run at the rate of more than 1K per second. The analog output channels, while capable of accepting digital samples at a maximum rate of 50K per second, was driven at the slower, 1K rate of the program. Maximum output resolution was 12 bits plus sign.

A set of 11 routines, all coded in assembly language, was required to implement the control functions. In Ref. 22, it was stated that 1227 words of core store were needed by the set. Using the characteristics published by SEL, they estimated that 943 to 1467 core store cycles were required per iteration, and 0.71 ms to 1.1 ms of computation time (exclusive of analog input delays) was also needed.

The J-85-13 engine does not represent an engine used in typical subsonic application, for it has added features such as a controllable variable geometry for the inlet duct, and a controllable exhaust nozzle area. We note that implementation in the computer of these two functions (they are two routines of the set of 11) required 297 words of core storage, 139 to 450 core store cycles, and 0.104 to 0.307 ms of computation time. These burdens were already included in the previously cited values for the set of 11 routines.

In summary, for the ATT application let us only deal with:

- Subsonic control functions
- Four-byte words (32 bits)
- An equivalent add time of 4.0 µs,

thus obtaining consistency with the previous estimation models. We thus obtain:

- Core storage requirement of 3720 bytes
- Maximum CPU utilization of 2.7 milliseconds per iteration per engine, or
- At 33 iterations per second, a CPU utilization of 89.1 milliseconds per second per engine, or
- For a four-engine aircraft, a CPU utilization of 356 milliseconds per second.

b. Some Interpretation of the Experiment and the Results

1) Duration of Computation

The computing load just estimated would appear to persist only during those periods, of tens of seconds duration, when the engines are undergoing major changes of operating state. Such changes will occur when (1) a change is made in the setting of the fuel control levers, (2) a change is made in the aircraft attitude or altitude, (3) air turbulence is encountered. These changes are prevalent during takeoffs, approaches, and landings.

During steady-state operation of the engines, hardly any changes to fuel flow should be required. Rather, the primary task of the engine control system could be one of monitoring those variables of the engine that are measured, together with other engine-sensitive variables (e.g., the settings of the fuel control levers), in order to detect any significant changes of value. Upon detection of that type of change, the control system should enter the elaborate control mode, reverting to the monitoring mode when steady-state engine conditions have been determined to exist once again.

2) Nature of the Computations

The calculation of the inlet duct variable geometry control, and of the compressor acceleration schedule, each used sets of stored curves and simple interpolation techniques. For the variable geometry application, three curves were stored, each for a particular value of compressor inlet temperature. A given curve provided the functional relationship between "variable geometry position" and corrected compressor rotor speed.

For the acceleration example, eight curves were stored, each for a particular value of compressor inlet temperature. These curves related rate of fuel flow to corrected compressor rotor speed.

It appears that these curves are empirically derived and are the same for all engines of a given type.

4. An Expanded Role for Computer Control of Engines

Mechanical and economic considerations limit the range of application for the hydromechanical controller. The limitation affects:

- The number of variables that can be handled.
- The complexity of the computations that can be performed.
- The ability to accommodate changes in the computational programs and stored constants.

For the ATT application, we might anticipate at least a doubling, likely a trebling, of the number of engine variables that are measured. Included in the set of added variables might be compressor temperature and pressure for both compressors, assuming a dual compressor engine, the rotating speed of the second compressor, exhaust gas temperature, engine exhaust pressure, and aircraft air speed.

It is unlikely that new engine parameters would be controlled; thus fuel flow and exhaust area might be the only controlled items.

A given type of engine may well have a variety of differing operating restrictions recommended for it. The sources of these restrictions may be the airline company, the engine manufacturer, and the FAA. Exceeding a given restriction need not result in engine failure; rather it may result in a shortening of the time to overhaul, or a shortening of the total life. With computer control the pilot need not be constrained, as now, to using one program of limits; rather, he may choose the one that best fits the situation. In normal instances, this may permit him to choose different programs during takeoff, cruising, and descent and landing. Such flexibility may actually result in more economic use of the aircraft.

In case of emergency, the pilot could override all constraints and drastically sacrifice engine life for overall safety of the aircraft.

Finally, it may well develop that normal trimming of the engines during maintenance may be facilitated by ready ability to adjust engine parameters in the program.

It is our estimate that the augmented control program would have the following expanded requirements:

- Core storage of about 5200 bytes (1300 32-bit words)
- A CPU utilization of 475 milliseconds per second for a four-engine aircraft, developed on the basis of 42,700 short operations and 19,000 long operations per second.

5. Reliability Considerations

a. Backup

Present practice in commercial aircraft does not provide for a backup of the hydromechanical controller. The primary reason is cost, and the use of multiengine aircraft. Since the failure of a controller affects only one engine, and since the probability of coincident failure of several controllers is considered acceptably low, this approach has found favor. In military fighter aircraft, however, a separate backup system is used; it is essentially a fuel valve that bypasses the controller. By monitoring the engine instruments, the fighter pilot is expected to control the fuel flow manually so that engine destruction is avoided prior to landing.

For the ATT application, we should assume that control of the engines is possible only via a computer, so that loss of all computational facility is equivalent to loss of engine control.

This is not to say that all features of engine control must be possible in the minimum case; but it is essential that at least the equivalent of manual control of the fuel valve should be possible. Even that seems too minimum.

b. Information Storage

Except for trim information for individual engines, all engine control program and data (e.g., curves) could be kept in an ROM. Indeed, each engine type could be served by one given control program and set of data. It is unclear as to how trim information need be or should be retained. Trimming is not a daily occurrence but takes place between extended periods. Thus, it seems unlikely that trim information (likely a couple tens of parameters per engine) might be integrated into the main ROM; on the other hand it need not be changed between all successive takeoffs. Loss of trim information should result in inefficient use of the engine, and not a hazard to safety.

Just how to enter (or reenter) trim information into the computer system is a minor question to be resolved, together with the question as to how that information can be retained during normal shut-down of the computer.

c. Checking Correctness of Operation

Because the engines have known limits on certain parameters (e.g., maximum exhaust gas temperature, maximum and minimum compressor speeds, maximum rates of compressor acceleration and deceleration), gross checks can be made of actual engine behavior, hence of the controlling program. These checks can be made by programs independent of the engine control program.

REFERENCES

1. Raymond C. Montgomery, "Analytical Design of Digital Flight Controllers," Presented at AIAA Conference on Guidance, Control and Flight Mechanics, April 16-18, 1971, Hempstead, New York.

2. Project 8274 final report.

3. "Low-wing-loading STOL Transport Ride Smoothing Feasibility Study--Final Report," No. D3-8514-2, The Boeing Company, Seattle, Washington (February 8, 1971).

4. "Analysis of Aeroelastic Mode Stability Augmentation Systems," No. D3-8390-4, The Boeing Company, Seattle, Washington (March 1971). In particular, the reader is directed to one paper (D3-8390-1) in that report, "Analysis and Mechanization of NASA-Langley Flutter SAS Concepts."

5. Microwave Journal, Vol. 15, No. 1 (January 1972). Alternative microwave landing systems are discussed in several articles in that issue.

6. RTCA, "A New Guidance System for Approach and Landing," SC-117, Document No. DO-148, December 18, 1970.

7. "Design, Development and Flight Evaluation of Inertially Augmented Automatic Landing Systems," Interim Report, Report No. ADR-754, Lear Siegler, Inc., p. III-2 (April 23, 1971).

8. T. Wempe and E. Palmer, "Pilot Performance with a Simulated Pictorial Landing Display Including Different Conditions of Resolution and Update Rate," Ames Research Center, NASA (Date unknown).

9. John D. Warner (Boeing), "Advanced Controls and Displays for Future Commercial Aircraft Operations," ATAA 2nd Aircraft Design and Operations Meeting, Los Angeles, California, July 20-22, 1970.

10. "L-193 Head-up Display (HUD) Manual," Librascope, Singer-General Precision, Inc. (date believed to be 1970).

11. "Mark 13 Area Navigation System," ARINC Project Paper No. 583, January 4, 1971.

12. ARINC Characteristic No. 561.

13. C. Broxmeyer, Inertial Navigation Systems, p. 133 (McGraw-Hill).

14. J. E. Volder, "The CORDIC Trigonometric Computing Technique," IRE Transactions on Electronic Computers, pp. 330-334 (September 1959).

15. E. R. Swanson, Navigation, Vol. 18, No. 2, pp. 168-175 (Summer 1971).

16. ARINC Characteristic No. 586.

17. "Technical Development Plan for a Discrete Address Beacon System (DABS)," Department of Transportation, FAA, Report No. FAA-RD-71-79 (October 1971).

18. "Aircraft Integrated Data System (AIDS Mark 2)," ARINC Characteristic 573-5, May 26, 1971.

19. "The Aircraft Gas Turbine Engine and Its Operation," Pratt & Whitney Aircraft (June 1952, rewritten Aug. 1970), Part No. PWA182408.

20. A. S. Boksenkom, "Dynamics and Control," Aircraft Propulsion, pp. 351-395, NASA, Proceedings of Conference at NASA Lewis Research Center, November 18-19, 1970, NASA SP-259.

21. "Digital Control of a Non-afterburning Turbo-jet," author and date unknown, document enclosed with letter by D. J. Arpasi, NASA Lewis Research Center, October 26, 1971, to R. Ratner.

22. D. J. Arpasi et al., "A General Purpose Digital System for On-line Control of Air-breathing Propulsion Systems," NASA TM X-2168, NASA Lewis Research Center (February 1971).

Appendix A

SOURCES OF INFORMATION ON WHICH ESTIMATES AND JUDGMENTS ARE BASED

The SRI project team had discussions either in person or by telephone with the following persons during the course of the work. These individuals, of course, bear no responsibility for the uses we have made of their information. We acknowledge here, with thanks, their cooperation and assistance.

There was available to us a larger number of individuals and organizations than the group we contacted. We selected those below to span reasonably the subject areas of interest, within the time and effort allocated to this task of the project.

ARINC, Washington, D.C.

R. Climie
D. H. Featherstone

Bendix Navigation and Control Division, Trenton, N.J.

E. Lademan
R. Belleman
G. Gartner
H. S. Dorman

Boeing, Seattle, Washington

R. E. Job
J. Small
H. Tobie
R. G. Young
J. Gannett
J. D. Warner
J. Headlund
F. Hall
J. Nesbitt
R. Williams

Collins Radio

L. R. Heron, San Mateo, Ca.
K. Engholm, Cedar Rapids, Iowa
Wm. Evans, Cedar Rapids, Iowa
N. B. Memesath

Delco Electronics, Milwaukee, Wisconsin

L. DeGroot
J. W. Sheldrick
L. D. Lewis
R. Farmer

Delta Airlines, Atlanta, Georgia

W. Overend

FAA, Washington, D.C.

T. Amlie
E. A. Post

Lear Siegler, Santa Monica, Ca.

H. Daubert
R. G. Gadbois
G. Moak

MIT Draper Laboratory

A. Engel
D. Keene
J. Barton

NASA Langley Research Center (in addition to R. Hood, N. Murray, W. Dove)

T. Walsh
J. Rainey
J. Bird
J. Hatfield
J. Painter

NASA Lewis Research Center

A. Boksenbom
D. Arpasi

NASA Ames Research Center

R. Bretoi
G. E. Cooper

Pan American World Airways, San Francisco, Ca.

I. Anderson
D. W. Frost
P. Jeffries, New York, N.Y.

Teledyne Systems Co.

E. Durbin

Trans World Airlines

R. W. Rummell, New York, N.Y.
R. L. Adams, Kansas City, Kansas
C. W. Carroll, Kansas City, Kansas

Appendix B

ILLUSTRATIVE DETAIL OF SUBSYSTEM ANALYSES OF THE ELECTRONIC DATA-PROCESSING SYSTEM

The following sections are included in this appendix:

(1) Area Navigation System (RNAV)

(2) Collision Avoidance System (CAS)

(3) Data Communications

(4) Graphic Display

(5) Inertial Navigation System.

1. Area Navigation System (RNAV)

Table B-1 gives the processing requirements for the Area Navigation System (RNAV).

2. Collision Avoidance System (CAS)

a. Processing Requirements

Table B-2 gives the processing requirements for the Collision Avoidance System.

Table B-1

PROCESSING REQUIREMENTS FOR AREA NAVIGATION SYSTEM (RNAV)*

Process	Iteration Rate	Prime Parameters	I/O Rates (No./Second)		Memory		Instrs./second		Time (percent)
			I's	O's	Instrs.	Data & Buffers	Shorts	Longs	
Horizontal Functions									
● RNAV Supervisor	--	Nav-system selection	--	--	75	15	--	--	--
● VOR/DME Navigation	5	Lat./long.	20	--	250	50	1,450	600	1.7
● Kalman Filtering	1/5	Lat./long.	1	1	250	65	150	50	0.2
● Alternative Nav.									
Mult. DME	5	Lat./long.	--	--	200	30	--	--	--
Hyperbolic	5		--	--	200	75	--	--	--
Air data	5	Smoothing of radio nav. data	--	--	110	25	--	--	--
● Processed flight	5	Distance-to-waypoint, time to-waypoint, cross-track deviation, automatic flight leg switching, etc.	--	65	450	100	10,000	4,400	11.0
● Airspeed (includes computation of temperature).	8	Mach number, knots	15	15	100	25	2,200	950	2.4
		Subtotals	35	80	1,635	385	13,800	6,000	15.3
Vertical Functions									
● Barometric alt. calibration and correction	16	h_b	16	16	30	5	560	160	0.5
● Baro-inertial filter	5	$h, \dot{h}$	10	10	80	15	300	30	0.2
● Profile (not automatic)	5	Deviation, pitch control	30	15	150	25	430	180	0.5
		Subtotals	56	41	260	45	1,290	370	1.2
Automatic Tuning	--	Communication and nav. units	--	--	200	50	--	--	--
		Totals	90	120	2,100	480	15,100	6,400	16.5

* These figures do not include the requirements for inertial systems, or for text and graphic display systems. These requirements are analyzed and tabulated in other sections of this report.

Table B-2

PROCESSING REQUIREMENTS--THE T/F COLLISION AVOIDANCE SYSTEM (CAS)

	Item		Function	Times per Second	Total Number of Operations per Second*									
					Detailed								Gross	
					F/S/B	A/S	M	D	Trigs		Trigs^{-1}	$\sqrt{\ }$	Shorts	Longs
									Small	Large				
	Co-Altitude Bands (Base and Predicted)		As function of own-altitude A and altitude rate $\dot{A}$ (ARINC 587, p. 72)	1	16	4	1	--	--	--	--	--	20	1
	Filter via $\dot{R}$		1) If $\dot{R} < (-\dot{R}_N$ knots), Exit 2) If $\dot{R} < (+80$ knots), Branch to R/A Evaluation	670	4000	1340	--	--	--	--	--	--	5340	--
	Filter via Altitude		1) If above or below, Exit 2) If within, encode (8 codes) • Co-altitude or • Non-co-altitude, but a) within predicted range b) Not within predicted range (ARINC 587, p. 72)	200	3200	1200	--	--	--	--	--	--	4400	--
	Filter via Tau Zone		1) If $[R > R_{02} + K\dot{R}]$, Exit 2) If $[R > R_{01} + K_1\dot{R}]$, encode $\tau 2$ If $[R \le R_{01} + K_1\dot{R}]$, encode $\tau 1$ (ARINC 587, p. 73)	20	200	80	40	--	--	--	--	--	280	40
	For Non-azimuth/Display Systems, Generate and Transmit Commands		Table look-up via altitude and τ_1/τ_2 codes (worst case--2 hazards; ARINC 587, pp. 76,77)	2/3	--	--	--	--	--	--	--	--	--	--
For Azimuth/Display Systems at 12 Hazards/Epoch (4/second)	Epoch Shift		$(R'_{n-6} \leftarrow R'_{n-5}),\ldots,(R'_{n-1} \leftarrow R'_n)$ $(\theta'_{n-6} \leftarrow \theta'_{n-5} + \Delta\theta_H),\ \ldots,\ (\theta'_{n-1} \leftarrow \theta'_n + \Delta\theta_H)$	4	104	28	--	--	--	--	--	--	132	--
	Smoothed R'_n, θ'_n		$R'_n = R''_n + \frac{(R_n - R''_n)}{2^k}$ (same for θ'_n)	4	16	24	--	--	--	--	--	--	40	--
	Projected R''_{n+1}, θ''_{n+1} (for next Epoch)		$\left.\begin{array}{l}\Delta R = (2R'_n - 3R'_{n-1} + R'_{n-2})\\ R''_{n+1} = R'_n + \Delta R\end{array}\right\}$ (same for θ)	4	40	32	8	--	--	--	--	--	72	8
	Epoch Interpolation		$K_e = (2000 - j)/2000$ $\Delta R_e = K_e(\Delta R);\ \Delta\theta_e = K_e(\Delta\theta)$	4	20	4	8	4	--	--	--	--	24	12
	Coordinate Conversion and Test for On-Scale [m = (n-6), 000n]		$X_m = K(R_m + \Delta R_e)\ \sin(\theta_m + \Delta\theta_e)$ $Y_m = K(R_m + \Delta R_e)\ \cos(\theta_m + \Delta\theta_e) - Y_d$	24	240	168	96	--	--	48	--	--	1128	480
	Trend Vectors ($1 \le i \le 5$) (and Test for On-Scale)		$\beta_n = \tan^{-1}(Y_n - Y_{n-1})/(X_n - X_{n-1})$ $\dot{\beta}_n = (2\beta_n - 3\beta_{n-1} + \beta_{n-2})/3$ (secs) $V = K\sqrt{(X_n - X_{n-1})^2 + (Y_n - Y_{n-1})^2}$	4	56	32	16	8	--	--	4	4	260	76
			$X_i = X_{i-1} + V\cdot\sin(\beta + \frac{2i-1}{2}\ \dot{\beta}K)$ $Y_i = Y_{i-1} + V\cdot\cos(\beta + \frac{2i-1}{2}\ \dot{\beta}K)$	20	120	220	120	--	--	40	--	--	940	440
	Aircraft Symbols (Triangular) (h = 1,2,3)		$\beta_A = \frac{\pi}{2} - \tan^{-1}(Y_{n+1} - Y_{n-1})/(X_{n+1} - X_{n-1})$	4	24	12	--	4	--	--	4	--	128	36
			$X_h = X_n + L_h\ \sin(\theta_h + \beta_A)$ $Y_h = Y_n + L_h\ \cos(\theta_h + \beta_A)$	12	48	48	24	--	--	24	--	--	456	216
	Hazard Data	Comp.	$[\tau = R/\dot{R}]$ or $[\text{CLOSE/OPEN}]_{\dot{R} < 80}$; $\Delta A = A_x - A_o$	4	40	12	4	--	--	--	--	--	52	4
		Location	Over/Below or Left/Right	4	80	10	--	--	--	--	--	--	90	--
	Host Aircraft Trend Vector and Pointer Symbol ($1 \le i \le 5$)		$X_i = X_{i-1} + V_g\cdot K\cdot\sin(\dot{\theta}_H \cdot \frac{2i-1}{2} \cdot K)$ $Y_i = Y_{i-1} + V_g\cdot K\cdot\cos(\dot{\theta}_H \cdot \frac{2i-1}{2} \cdot K)$	5	30	40	35	--	5	5	--	--	210	105
			$\beta_h = \tan^{-1}(X_5 - X_4)/(Y_5 - Y_4)$	1	4	2	--	1	--	--	1	--	29	8
			$X_h = X_5 + L_h\ \sin\ \beta_h$ $Y_h = Y_5 + L_h\ \cos\ \beta_h$	3	12	6	6	--	--	6	--	--	108	54
	Track and North Data Computation and Symbol Generation		$X_T = K\ \sin(\psi - \theta_H);\ Y_T = K\ \cos(\psi - \theta_H) - Y_D$ $X_N = K\ \sin(\theta_H) + X_D;\ \ Y_N = K\ \cos(\theta_H) + Y_D$	1	8	7	4	--	--	4	--	--	75	36
			$X_h = X_N + L_h\ \sin(-\theta_H)$ $Y_h = Y_N + L_h\ \cos(\theta_H)$	3	12	6	6	--	--	6	--	--	108	54
			TOTALS		8270	3275	368	17	5	133	9	4	13892	1570

*The column heading abbreviations are:

F/S/B	--Fetch/Store/Branch
A/S	--Add/Subtract
M	--Multiply
D	--Divide
Trigs, Trigs^{-1}	--Trigonometric Evaluations
$\sqrt{\ }$	--Square Root
Shorts	--Instructions F/S/B, A/S
Longs	--Instructions M, D

Notes:

1. Plots and data for all major threat aircraft would most probably be presented in red color, and/or in an intensified and/or blinking mode. Aircraft-classified Tau-1 and those at short ranges would be in this category.
2. The presentation would be reconstituted every 3 seconds. The aircraft symbols (triangles) would progress along the trend vectors.

b. Memory Requirements of the T/F Collision Avoidance System (CAS)

Function	Program	Data and Buffers
Co-altitude Bands Determination	30	
Filter via $\dot{R}$	15	
Filter via Altitude	25	
Filter via Tau. Zone	30	
Generation of Commands (Nondisplay)	20	3
Display Operations (12 Max. Hazard Aircraft)		
Epoch-shift of R/θ Data	25	72
Smoothed R_n', θ_n'	20	12
Projected R''_{n+1}, θ''_{n+1}	20	12
Epoch Interpolation	15	
Coordinate Conversion and Test	25	84
Hazard Aircraft Trend Points	50	60
Generation of Aircraft Symbols	20	36
Hazard Data	40	36
Host Aircraft Rend Vector and Symbol	35	8
Track and North Computation	30	2
Range Circles, Track Scales, etc.	50	275
Totals	450	600

c. Basic Assumptions Made for Analysis of T/F Collision Avoidance System

1) Number of Slots with Data to be Processed

- Of the 2000 slots, worst-case traffic conditions will generate data in 1000 slots.
- Of the 2000 data sets processed, 600 require processing beyond the range-rate, 60 beyond "co-altitude," and 12 through complete processing for CRT display (the non-CRT system accommodates a maximum of only two in generation of an evasive command).

2) Basic Processing Organization

The SRI architectural philosophy will probably dictate against an interrupt-mode of processing such as would be involved if the data of each 1.5-ms slot were to be processed immediately after the end of that slot. It is therefore assumed that dual buffers are provided for storage of N sets of data (each), so that the computer is directed for CAS computations only every 1.5N ms. At 2 packed words of 32 bits each per slot, 400 words of buffer would accommodate a processing period of 150 ms.[*]

3) Auxiliary Operations

It is assumed that the (central) computer is not responsible for the following miscellaneous operations, these being performed by the CAS black box:

- Slot acquisition.
- Synchronization of the clock via ground stations (or via other aircraft when not within range of ground stations).
- Determination and use of "Hierarchy"--a number system designating relative accuracy of current synchronization, eligibility to synchronize other aircraft, etc.
- Formatting of "own-slot" transmissions, and decoding of R, $\dot{R}$, and altitude of "other-slot" receptions.
- Antenna switching and frequency switching.
- Shift to Back-up Mode.
- And so on.

[*] This is a conservative allotment. If it develops that the computer can process the N data sets very rapidly, only a small second buffer will be required for storage of new data accessed during the "current" processing period, the main buffer then becoming available again.

4) Azimuth and CRT Display Capability

As noted above, an azimuth/display capability is assumed for the EDP-requirement projections of this report because of the potential for increased CAS effectiveness--particularly for terminal-area operations.

3. Data Communications

Table B-3 gives the processing requirements for data communications.

4. Graphic Display

a. Processing Requirements

Table B-4 gives the processing requirements for graphic display.

b. Memory Requirements of a Graphic Display System

1) The Basic Programs

Function	Program	Data and Buffers
Interrupt or Polling Routines for Keyboard Processing	80	--
Sector Selection (from 40 sectors at 100 miles each)	25	5
Rough Selection of Points λp, φp	30	10
Precision Computation of ρ, $(\theta_H + \theta)$	70	20
Conversion of Coordinates to X, Y and Off-scale Tests	25	10
Add ΔX, ΔY for Circular Symbols	15	--
For lines (or polygons) going off-scale, compute coordinates of border intersections	25	5
Designation of Name/Freq./Alt. Locations	20	5
Computation of Curved Trend Vectors	40	5
Computation of Track Data and Symbols	25	5
Buffer Storage of Display Data: Variable	--	110
Fixed*	10	35

* Fixed elements are the border, the map side, the aircraft symbol, range circles, extended heading line, and portions of a linear-scale track presentation at the top of the display.

Table B-3

PROCESSING REQUIREMENTS FOR DATA COMMUNICATIONS

	Function	Iteration Rate (no./s)	Data Transfers (no./s)		Processor Requirements		Memory	
			I	O	Instr. (no./s)	Time (percent)	Program	Buffer
Internal	Aircraft Integrated Data Systems (AIDS)	1/4,1,4	100	200	50	0.02	30	*
	Airspeed (including temperature)	8,16	32	--	240	0.10	20	*
	Autopilot/Autoland	160,20	820	[--]	1700	0.68	20	*
	Automatic Tuning (Nav/Com)	N	N	N	N	N	10	*
	Collision Avoidance (CAS)	6	1400	225	60	0.02	10	400
	Discrete Parameters	1,5	5	5	60	0.02	20	*
	Displays: Graphics	10	23	160	300	0.12	30	*
	Text (CDU)	10	N	280				
	Electronic Attitude (EADI)	10	--	700				
	Dynamics: Flutter Control	250	(+500)	(+0)	(2500)	--	10	*
	Stabil Augment	240	2400	1680	2500	1.0	10	
	Engine Control	30	1440	240	300	0.12	10	*
	Navigation: Crosstrack Dev., Distance/Time, etc.	5	--	--	N	N	10	*
	Dual VOR/DME	5	20	--	50	0.02	10	*
	Inertial	25,5	200	5	300	0.12	10	*
	Kalman Filtering	1/5	[1]	[1]	--	--	--	--
	Instruments/Indicators	10	--	200	100	0.04	10	*
	Subtotals		6440	3695	5660	2.3	210	400*
Air-Ground-Air	Terminal Area Messages (ATC, IPC, etc.)	2	12	6	500	0.2	250	12
	Enroute ATC and Company Messages	N	N	N	N	N	200	100
	Address Decoding of All Interrogations (Burst Rate)	800	<800>	--	<4800>	<1.9>	[100]	*
	Subtotals		12	6	500	0.2	450	112
	TOTALS		6450	3700	6160	2.5	660	510

Notes:
- * -- Common buffer used.
- ( ) -- Processing not required during approach/landing--the worst-case situation.
- [ ] -- Transfers or memory common to other function(s).
- N -- Negligible requirements.
- < > -- Address decoding performed by special hardware (to preclude the need for high-rate polling or interrupts.

1) The Basic Programs (continued)

Function	Program	Data and Buffers
Data-output Routine	10	--
Computation of Translation/Rotation at (8 - 1 = 7) intermediate times	15	5
Slewing Display to "Phantom" Displaced Reference	50	5
Additional Programs for North-oriented Display	50	10
Additional Vertical-profile Display Programs	200	30
Subtotals	690	260

2) Horizontal Display Parameters

125 VOR/DMEs at 6 Words each (precision lat./long.)	750 Words
75 Intersections and Waypoints at 8 (precision lat./long.)	600
15 Instrument Landing Systems at 24 (precision lat./long.; includes markers, runways, extended center line, names)	360
30 SIDS/STARS/MAPS* at 40	1200
200 Airway Designators† at 2	400
10 Prohibited and Defense Areas, etc. at 30	300
200 Points of lat./long. Grid at 1	200
200 Lat./Long. Designators† at 2	400
1 Track/Bearing Scale at 30	30
Symbols	250
2 Circles at 32 = 64	
2 Hold Patterns at 40 = 80	
6 Symbols at 10 = 60	
6 Symbols at 4 = 24	
4 Vectors at 5 = 20	
Subtotal	4500 Words

* Standard Instrument Departures, Standard Terminal Arrivals/Missed Approach Procedures--data regarding courses, altitudes, etc.

† All designators (e.g., 25W, 70N, V259) assumed to be displayed horizontally and with no processing against overlay with other symbols.

3) Vertical Profile Modes--Additional Storage Requirements

16 Grid Segments and Coordinates at 3 Words each	50
20 Profiles at 25	500
5 Special Symbols at 10	50
Subtotal	600

5. Inertial Navigation Systems

a. Processing Requirements

Table B-5 gives the processing requirements of an inertial navigation system.

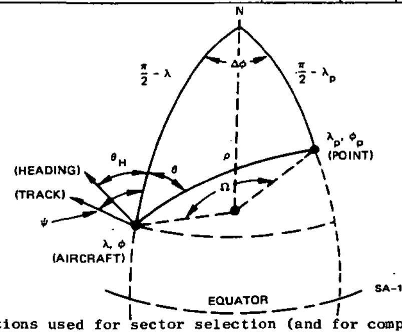
b. Memory Requirements of an Inertial Navigation System

Function	Program	Data and Buffers
Interrupt Routines for Mode and Keyboard Operations	100	--
"Inner Loop" (at 25/s)--Steps 1-6	121	31
Lat./Long. and $V_N/V_E/V_Z$--Steps 7-13	145	15
Misc. Parameters--Steps 14-19	127	15
Waypoint Calculations (and Storage)	91	30
Binary/Decimal Conversion and Output	30	2
Timing Interrupts--Step 22	30	--
Gimbal Drive--Step 23	300	25
System Initialization (Pre-flite only)	750	20
Execution and Self Check (not included here)	--	--
"Inverse" λ, Ø,V,C_S Computations for "Polar" Navigation	90	10
Subtotals	1800	150

Table B-4

PROCESSING REQUIREMENTS FOR TRACK-ORIENTED HORIZONTAL (GRAPHIC) DISPLAY

Item	Function	Times per Display at 1/s	Total Number of Operations per Computer Display (at 1/sec)								
			Detailed							Gross	
			F/S/B	A/B	M	D	Trigs.		Trigs^{-1}	Shorts	Longs
							Small	Large			
Sector Selection	Designate occupied sector as f(dist. from origin).* Additional sectors (±) as f(scale).	1	38	25	6	3	3	5	3	242	91
Rough Selection of Points λ_p, φ_p	Yes if: $K\left[\Delta\lambda + \frac{\Delta\varphi}{\cos\lambda}\right] < \left\{\begin{matrix}\text{Ahead--2.5 } Y_{max} \\ \text{Behind--1.7 } Y_{max}\end{matrix}\right\}$	25	200	150	25	--	--	1	--	365	33
Reference-Pt. Comps Angles of "Convenience" α	$\tan^{-1}\left[\frac{1}{\tan_s(\Delta\varphi/2)} \cdot \frac{\cos(\lambda p-\lambda)/2}{\cos(\lambda p+\lambda)/2}\right]$	25	225	175	25	25	25	50	25	2050	800
β	$\tan^{-1}\left[\frac{1}{\tan_s(\Delta\varphi/2)} \cdot \frac{\sin_s(\lambda p-\lambda)/2}{\sin(\lambda p+\lambda)/2}\right]$		225	125	25	25	25	25	25	1700	600
Earth Angle (Ω)	$2\tan^{-1}\left[\tan_s\left(\frac{\lambda p-\lambda}{2}\right)\frac{\sin(\alpha)}{\sin(\beta)}\right]$		75	75	25	25	25	50	25	1800	800
Distance (ρ)	$(KR_e)\ \Omega$ (Deflection Units)		25	--	25	--	--	--	--	25	25
Bearing $(\theta_H+\theta)$	$\theta_H + \sin^{-1}\left[\frac{\sin_s(\Delta\varphi) \cdot \cos(\lambda p)}{\sin_s(\Omega)}\right]$		175	25	25	25	50	25	25	1200	550
Reference-Pt. Comps and Off-scale Tests	$X_R = \rho \cdot \sin(\theta_H + \theta)$ $Y_R = \rho \cdot \cos(\theta_H + \theta) - Yd$ (Packed)	25	300	200	50	--	--	50	--	1250	450
Slave-Point Comps and Off-scale Tests (100 points)	$X_S = X_R + \Delta X \cdot \cos(\theta_H) + \Delta Y \cdot \sin(\theta_H)$ $Y_S = Y_R - \Delta X \cdot \sin(\theta_H) + \Delta Y \cdot \cos(\theta_H)$	100	2200	600	400	--	--	2	--	2830	416
Add ΔX, ΔY for Circular Symbols	$X = X_o + \Delta X$ $Y = Y_o + \Delta Y$ (Packed)	8	1024	512	--	--	--	--	--	1536	--
For lines going off-scale, comp. Border intersection	e.g. $y'_{p+1} = y_p + (y_p - y_{p+1})\left(\frac{X_p - X_{max}}{X_{p+1} - X_p}\right)$	10	130	80	10	10	--	--	--	210	20
Name/Freq/Alt Location	Over/below or left/right positioning re item ref. point	10	160	20	--	--	--	--	--	180	--
Generation of Curved Trend Vectors $(1 \le i \le 5)$	$X_i = X_{i-1} + V_g \cdot K \cdot \sin_s\left[\dot{\theta}_H\left(\frac{2i-1}{2}\right)K\right]$ $Y_i = Y_{i-1} + V_g \cdot K \cdot \cos\left[\dot{\theta}_H\left(\frac{2i-1}{2}\right)K\right]$	5	20	20	62	1	5	5	--	180	70
Track and North Data Computation and Symbol Gen.	$X_T = K\sin(\psi - \theta_H)$; $Y_r = K\cos(\psi - \varphi_H) - Y_D$ $X_N = K\sin(\theta_H) + X_D$; $Y_N = K\cos(\theta_H) + Y_D$	1	30	20	5	--	--	4	--	79	37
Translation/ Rotation	$\Delta Y = K \cdot V_{ground}$ $\Delta\theta = (\theta_H)_n - (\theta_H)_{n-1}$	7	28	7	7	--	--	--	--	35	7
Data Output	(DMA) (Included in data communication)	8	--	--	--	--	--	--	--	--	--
	TOTAL		4855	2034	690	114	133	217	103	13682	3899



GREAT-CIRCLE TRIGONOMETRIC RELATIONSHIPS

*The spherical trigonometric calculations used for sector selection (and for computation of the "phantom" aircraft location for slewing) are essentially the same as those used in the precision computations for the points λ_p, φ_p. (Assumed here that slewing is not used during approach.)

Table B-5

PROCESSING REQUIREMENTS OF AN INERTIAL NAVIGATION SYSTEM

Step	Item	Function	Item (Times /s)	Total Number of Operations Per Second									
				Detailed								Gross	
				F/S/B	A/S	M	D	Trigs. Small	Trigs. Large	Trig^{-1}	$\sqrt{\ }$	Shorts	Longs
1	R_{sx}, R_{sy}, R_{sz} ($\dot{R}_s$ ditto)	$(R_e+h)\begin{bmatrix}\cos\lambda g\cdot\cos(\delta t+\varphi)\\ \cos\lambda g\cdot\sin(\delta t+\varphi)\\ \sin\lambda g\end{bmatrix}$ ($\dot{R}$ ditto)	25	425	50	200	--	--	100	--	--	1975	1000
2	$\Delta\begin{bmatrix}U\\V\\W\end{bmatrix}$	$\begin{bmatrix}R_{sy}f_{sz} - R_{sz}f_{sy}\\ R_{sy}f_{sx} - R_{sx}f_{sz}\\ R_{sx}f_{sy} - R_{sy}f_{sx}\end{bmatrix}$	25	300	75	150	--	--	--	--	--	375	150
3	$\int\Delta\begin{bmatrix}U\\V\\W\end{bmatrix}$	$\begin{bmatrix}U + \Delta U\\ V + \Delta V\\ W + \Delta W\end{bmatrix}$	25	150	75	--	--	--	--	--	--	225	--
4	$R\dot{R}$	$R_{sx}\dot{R}_{sx} + R_{sy}\dot{R}_{sy} + R_{sz}\dot{R}_{sz}$	25	150	50	75	--	--	--	--	--	200	75
5	$\Delta\begin{bmatrix}R_{sx}\\R_{sy}\\R_{sz}\end{bmatrix}$	$\frac{1}{R^2}\begin{bmatrix}0 & -R_{sz} & R_{sy} & R_{sx}\\ R_{sz} & 0 & -R_{sx} & R_{sy}\\ -R_{sy} & R_{sx} & 0 & R_{sz}\end{bmatrix}\begin{bmatrix}U\\V\\W\\R\dot{R}\end{bmatrix}$	25	500	150	425	25	--	--	--	--	650	450
6	$\int\Delta\begin{bmatrix}R_{sx}\\R_{sy}\\R_{sz}\end{bmatrix}$	$\begin{bmatrix}R_{sx} + \Delta R_{sx}\\ R_{sy} + \Delta R_{sy}\\ R_{sz} + \Delta R_{sz}\end{bmatrix}$	25	150	75	--	--	--	--	--	--	225	--
7	R_e	$\begin{bmatrix}R_{sx}\cos(\delta t) + R_{sy}\sin(\delta t)\\ \text{Ditto (form)}\\ \text{Ditto}\end{bmatrix}$	5	70	15	40	10	--	--	--	--	85	50
8	$\dot{R}_e$	$\begin{bmatrix}(\dot{R}_{sx} + \delta R_{sy})\cos(\delta t) + (R_{sy} - \delta R_{sx})\sin(\delta t)\\ \text{Ditto (form)}\\ \text{Ditto}\end{bmatrix}$	5	90	45	60	--	--	--	--	--	135	60
9	Variables of Convenience R_k	$R_{ex}^2 + R_{ey}^2$	5	20	5	10	--	--	--	--	--	25	10
10	R_j	$\sqrt{R_k}$	5	--	--	--	--	--	--	--	5	100	25
11	Latitude λ	$\sin^{-1}\left[\frac{R_{ez}}{R}\left(1 + \frac{2c\,R_k}{R^2}\right)\right]$	5	20	10	15	10	--	--	5	--	145	65
12	Longitude φ	$\cos^{-1}\left[\frac{R_{ex}}{R_j}\right]$	5	10	--	--	5	--	--	5	--	125	40
13	Velocities (V_N) (V_E) (V_Z)	$(R\dot{R}_{ez} - R_{ez}\dot{R})/R_j$ Ditto (form)	5	60	15	30	15	--	--	--	--	75	45
14	C_s^t	$\begin{bmatrix}-\sin\lambda\cdot\cos(\delta t+\varphi) & -\sin(\delta t+\varphi) & -\cos\lambda\cdot\cos(\delta t+\varphi)\\ -\sin\lambda\cdot\sin(\delta t+\varphi) & \cos(\delta t+\varphi) & -\cos\lambda\cdot\sin(\delta t+\varphi)\\ \cos\lambda & 0 & -\sin\lambda\end{bmatrix}$	5	90	10	30	--	--	20	--	--	400	190
15	Pitch Roll Heading	f (C_s^t, Gimbal Resolvers)	5	300	225	150	--	--	30	--	--	975	390
16	Track (Ψ)	$\tan^{-1}(V_E/V_N) + K$	5	50	25	--	5	--	--	5	--	190	45
17	Ground Speed (V_g)	$\sqrt{V_N^2 + V_E^2}$	5	20	5	10	--	--	--	--	5	125	35
18	Wind Components (V_{WN}) (V_{WE})	$V_{AS}\cdot\cos\theta_H - V_N$ $V_{AS}\cdot\sin\theta_H - V_E$	1	4	2	2	--	--	2	--	--	21	18
19	Drift Angle (θ_d)	θ - Ψ	1	2	1	--	--	--	--	--	--	3	--
20	Waypoint Calculations												
	Angles of Convenience α	$\tan^{-1}\left[\frac{1}{\tan_s(\Delta d/2)}\cdot\frac{\cos(\lambda p-\lambda)/2}{\cos(\lambda p+\lambda)/2}\right]$	2	18	14	2	2	2	4	2	--	164	62
	β	$\tan^{-1}\left[\frac{1}{\tan_s(\Delta d/2)}\cdot\frac{\sin_s(\lambda p-\lambda)/2}{\cos(\lambda p+\lambda)/2}\right]$		18	14	2	2	2	2	2	--	138	48
	Earth Angle (Ω)	$2\tan^{-1}\left[\tan_s\left(\frac{\lambda p-\lambda}{2}\right)\cdot\frac{\sin\alpha}{\sin\beta}\right]$		12	6	2	2	2	4	2	--	150	64
	Distance (ρ)	$(K\cdot R_e)\,\Omega$ (Nautical Miles)		4	--	2	--	--	--	--	--	4	2
	Bearing (θ)	$\sin^{-1}\left[\frac{\sin_s(\Delta d)\cdot\cos(\lambda p)}{\sin_s(\Omega)}\right]\cdot K$		14	2	4	2	4	2	2	--	144	62
	Time (τ)	ρ/V_g (Hrs, Mins, Secs)		4	--	2	2	--	--	--	--	4	4
21	Binary/Decimal Conv. & Output	(For Control-Unit Display)	2	30	10	20	--	--	--	--	--	40	20
22	Timing Interrupts	(From 25/s, 5/s, 2/s counters)	25	500	250	--	--	--	--	--	--	750	--
23	Gimbal Drive	f(C_s, Gyros)	25	1500	1125	750	--	--	150	--	--	4875	1950
		TOTAL		4511	2254	1981	80	10	314	23	10	12323	4860

PART 2

TECHNOLOGY

TABLE OF CONTENTS FOR PART 2

LIST OF TABLES AND ILLUSTRATIONS

I INTRODUCTION

The purpose of the technology study is to identify the most appropriate technologies to be used in ultrareliable airborne computers. The time span for product development and product construction is 1973-1975, with the requirement that the chosen technology lends itself to full production at the end of the period. The computers thus designed will be used in aircraft in operation 1975-1985.

The technology study establishes pertinent characteristics such as general technical feasibility, environmental limitations, speed of operation, cost of development, cost of production, reliability, and maintenance.

The report is divided into three parts: logic circuits in processor and subsystems, memory technology, and interconnection technology.

II LOGIC CIRCUITS IN PROCESSOR AND SUBSYSTEMS

A substantial portion of the logic in equipment expected to be developed the next few years will be based on Large Scale Integration (LSI) techniques. Whether standard off-the-shelf items or customized designs are used depends on expected volumes of production and the complexity of the desired logic functions. The conventional ICs, Small Scale Integration (SSI), represent an integration only on the component and device level, while the interconnection between elementary functions, such as gates and flip-flops, must be made outside the circuit. The utilization of SSI for logic design has, in the last few years, resulted in substantial cost and volume savings. The separation of the elementary functions on the ICs has made standardization feasible and production costs have been drastically reduced. The functional design of more complex functions has been made by interconnecting gates, flip-flops and other circuits through traces on printed circuit boards, through wire wrapping and other means. The costs associated with wiring, both labor and material, has provided the incentive to take the next step in circuit integration. A substantial reduction in interconnections between elementary functions could be made inside the integrated circuit. Another benefit of this would also be an increase in the gate-to-pin ratio of the integrated circuit. The process technology had advanced to a point where instead of 10-20 gates, as in SSI, up toward 100 gates could be placed on a chip without prohibitive reduction in production yield.

Now the big question was how to take advantage of the LSI capability. The cost of designing a complex integrated circuit was substantial and is still in the order of $20,000 per customized circuit. Only high volume production could justify this initial expense. The semiconductor manufacturers were searching for commonly used complex functions. The result of this search was the line of Medium Scale Integrated Circuits (MSI). The most attractive MSI circuit is the shift register, in that a very large number of stages can be interconnected with very few leads leaving the circuit and very efficient circuit layout. For example, the list of MSI circuits available from Texas Instrument includes counters, decoders, data selectors/ multiplexers code converters. A number of arithmetic elements for up to

4-bit parallel operations are available, supplemented with 4-8 bit registers of different kinds. Random and read-only memories (RAMs and ROMs) also belong to the MSI list. Most available MSI circuits are based on transistor-transistor logic (TTL) but shift registers, RAMs, ROMs, and a few other circuits are also available in metal oxide semiconductor (MOS) technique.

Increased yields and increased circuit densities possible in TTL circuits (and even more so in MOS circuits) would make it desirable to design more complex standard circuits. We see that, with the exception of memory functions, the MSI circuits are equivalent to, at most, 50-60 gates. The search for commonly used circuits with higher complexity than 50 gates, with the exception of memory functions, give very small return. The arithmetic elements, for instance, offer some further possibility in increasing the number of bits or in including not only the arithmetic function but also one or several registers in a vertical slice through an arithmetic unit. Increased number of bits above 4 results in decreasing number of users, as the arithmetic circuits come typically modulo 4 (bits), some modulo 8, but decreasing numbers for other bit counts. The number of leads to the package tends to increase linearly with the number of bits in the arithmetic element, reducing the advantage of increased integration. The bit slice approach has a passing advantage, as it assumes a traditional computer organization which no longer is accepted widely enough.

The difficulties in using standard LSI in the structured arithmetic portion of a computer are far exceeded by the difficulties in utilizing standard LSI for less structured logic, such as is used in the control logic of the processor or in the logic for subsystems such as I/O equipment, etc.

The logic designer, in order to take advantage of the compactness and reliability of LSI, must resort to customized design (pay for the cost of developing the circuit specifically for his needs). If the volume of equipment to be produced is high, as in a desk calculator, an ideal situation exists. Most other applications, however, provide a less clear-cut choice. Before looking at the alternatives available for the small volume producer, let us dwell for a moment on the desk calculator development.

The desk calculator, being a mass-produced item containing a modest amount of logic, was a natural candidate for LSI. A gradual replacement of discrete components and SSIs evolved. There were minor differences in the design approach taken by the different manufacturers that had to be accommodated for, resulting in great difficulties in standardizing circuits for general use. The number of LSI chips per calculator, based on the MOS technology, increased first when more and more of the previously used circuits were replaced. Concurrently, the advances in the MOS technology increased the number of circuits per chip, with the effect of again reducing the number of chips from a peak of 5-7 chips down to presently one chip only for almost all of the circuits for the simple calculator. Some attempts were made in standardization. One manufacturer offered a set of 6 chips for a complete 8-digit calculator. We do not know how successful this approach has been, but the availability of a built-in Read-Only Memory (ROM) for function control should provide some flexibility in responding the special custom requirements. The read-only memory or programmable logic approach is now incorporated in several designs standardized to meet a variety of customer demands.

The calculator development has now reached a point where the calculator manufacturer is less concerned about the logic refinement inside the LSI chip, than with his ability to satisfy the special operational and human engineering aspects of design.

The calculator on a chip is now an accepted fact; the cost per chip is subject to strong competition, with chip costs in the $20-$30 range now and expected to go below $10 in 1974.

Some important parallels can be drawn between the calculator development and the development of LSI oriented minicomputers, but there are also some big differences. The minicomputers all have different approaches in the logic structure, differences that have made it difficult to offer standard products acceptable to a broad market. The byte slice approach has had some acceptance but, as mentioned earlier, is of passing value, as more circuits can be placed on a chip. Any real breakthrough will not take place before complete CPUs can be offered as components to the application-oriented system and subsystem manufacturer. Some LSI calculator chips have the potential for expansion into the lowest end of the computer market, and are advertized as computers. Bit serial BCD operation at clock frequencies typically at 100-200 kHz limits the usefulness of these chips to low-speed applications.

Probably the most advanced attempt in making an LSI computer is the system IV/70 from Four-Phase Systems Inc. This computer acts as a communication display processor interfacing up to 32 display terminals, a central computer direct or remote, and several peripheral equipments. The computer consists of 12 LSI chips (MOS), 6 of which are 48,000 bits of ROM, 3 arbitrary logic and 3 byte-sliced arithmetic chips. This computer was planned around 1968-69 and is just now going into production. The extensive use of ROMs for control of the 120 machine instructions should provide a high degree of freedom in modification and adaptation to other applications.

Microprogramming, as in the example above, is one of the means available to reduce the development cost of computers. The production cost of systems using ROMs instead of arbitrary logic on customized LSI is higher but is still justified for the following reasons:

1) Initial development cost is significantly lower.

2) Design modifications are simpler with lower cost and turn-around time.

3) The savings from 1) and 2) on low volume production will offset the higher production cost.

Cost per bit of ROMs is continually reduced, resulting in improving economies, but other means of reducing the cost of the control memory are sought. One method of reducing the number of control bits is to replace the ROM with a smaller RAM, backed up with some slower non-volatile memory. In this organization, the control memory stores only the control information to be used in a given mission. At change of mission, the control memory is changed to satisfy the new needs. This approach is especially satisfactory in developing complex systems where many modifications can be expected, both in the debugging phase and later in the use of the system. Another motivation for using this scheme is that it is economically feasible to use high-speed bipolar RAMs, rather than the slower MOS ROMs. Bipolar ROMs represent a compromise between the two.

An alternative is also the usage of programmable read-only memories (PROMs). These control memories are of great help in the debugging state. If replaced with regular, less costly, ROMs for the production units, redesign of the ROM patterns may still be required for adaptation to new applications.

All control memory approaches, whether they are ROMs, RAMs, or PROMs are essentially listings of control bit patterns. Different schemes are used to reduce the number of control words. The address of the next word in a control cycle is part of the output word. Combinations of the address portion of the control word operation code and state signals are also used for pointing to the address of the succeeding control word.

The Programmable Logic Approach (PLA) is different in that the control signals are formed as the logic output of a gate structure in a similar way to customized arbitrary logic. The difference from customized random logic is that the development cost is significantly lower and design modifications can be made with relative ease. In one approach, the programming is made by changing the top layer metal interconnect pattern or the oxide cut mask, in order to connect or disconnect gate inputs. The similarity in cost of design and redesign to ROMs is evident. The programmable logic approach tends to be more economic than ROMs when the number of control bits exceeds a certain number. One commercially available PLA has 17 external inputs, 18 outputs representing sums of up to 60 product terms formed by the 17 inputs and the outputs from 8 state flip-flops, also located on the chip. The inputs to the flip-flops come from 16 sum-of-product terms generated on the chip, in addition to the 18 available at the output.

This type of programmable logic array operates at clock rates of 200 kHz. The speed limitation is due to the high capacitive loads in the high fan-in gates. Improved device technology can, of course, improve the speed by almost an order of magnitude. The speed should, however, not be directly compared to the cycle time of an ROM, as somewhat more time-preserving design approaches can be taken with PLAs.

Micro-cellular programmable logic arrays, where only a limited number of cell inputs are connected to a given cell output, can operate at substantially higher clock rates. This higher cell speed is, however, offset to some degree by the fact that several cell delays are accumulated in each logic decision.

Both types of PLAs can be customized by programming a single mask pattern. The arrays can also potentially be programmed electronically as the PROMs or by including control flip-flops or other memory bits on the chip that can be preset either by shifting a signal pattern or by some form of memory write process. The electronically programmable array has all the advantages of low hardware development cost, adaptability to new application and mission requirements, and low production cost, due to the high volume. There would certainly be great hesitance among persons responsible for design and use of airborne computers, if the logic performance is dependent on volatile memories. On the other hand, as will be further discussed in the memory section, volatility is a relative subject in that power can be provided during long power interruption by batteries supplying at least the function control memories, and in all probability also the active logic portion of the CPU circuits.

Cost Comparisons - Logic

In this section, we will attempt to compare the cost in designing and producing logic circuits utilizing the different concepts discussed in the previous section. The purpose is not to establish absolute cost comparisons but rather, to show trends in cost relations as function of production volume.

Table 1 shows the logic cost for the different techniques. ROMs, PROMs, and RAMs are assumed to be used as microprogram stores--a range of 10 to 20 bits of memory was assumed to be equivalent to one logic gate. The number of gates that can be replaced by the Programmable Logic Array from Texas Instruments is somewhat uncertain and certainly application dependent. Access and cycle times plus gate delays are also shown, in order to give the proper indication of cost/performance.

Figure 1 shows the cost per gate as a function of the number of units used for the different design and production techniques. A unit here is defined as a packaged circuit containing a number of gates or storage bits in a fixed combination. A conventional SSI circuit of a certain type (e.g., quadruple 2 input NAND) is the least complex unit produced in very large quantities and is used in relatively large numbers. The circuit cost shown for SSI in the diagram includes cost for assembly, material, and labor and takes into account cost variation as function of volume of purchase and production. An initial development cost, after logic design, is taken into account for all categories. This may be as low as $.05-$1.00 per gate, mainly for the PC board layout, when using MSI, and up to $200/gate for customized bipolar LSI. Programmable ROMs and RAMs as used for microprograms have no visible initial development cost for the user. De facto, some costs not shown in the diagram should be added for the in-house conversion of the logic equations to program code in the memories. In order to compare alternative approaches to realizing complex logic, curves for memory arrays are given based on a statistical equivalence between a set of bits in a control memory and a gate in an arbitrary control network. Most of the curves assume an equivalence of 20 bits per gate, but one pair of curves are given for 10 bits per gate.

Some caution should be expressed in using the diagram in comparing low complexity circuits with the high complexity ones. One single system may use, say, 10 MOS LSI circuits with each circuit of a different design and the total system may be equivalent to 1000 gates. If 100 systems are built,

Table 1

Cost Comparisons (Early 1972)

	Bits/Chip	Equivalent Gates/Chip	Cost/Bit *100/10,000* quantities	Cost per Gate Equivalent*		Development Cost†	Access Time Gate Delay ns	Cycle Time ns
				100 quantities	1000 quantities			
Read Only Memories								
Bipolar	1024	50-100	1.2/.8¢	12-24¢	8-16¢	6-12	45	45
MOS - Static	2048	100-200	.8/.4¢	8-16¢	4-8¢	3-6	900	900
Dynamic	2048	100-200	.8/.4¢	8-16¢	4-8¢	3-6	500	500
Programmable ROMs								
Bipolar	256	13-26	5/1.5¢	50-100¢	15-30¢	1-2	50	50
MOS - Static	2048	100-200	4/2¢	40-80¢	20-40¢	--	1000	1000
Dynamic	2048	100-200	4/2¢	40-80¢	20-40¢	--	650	650
Random Access Memories								
Bipolar	256	13-26	10/5¢	100-200¢	50-100¢	--	100	100
MOS - Static	256	13-26	5/2.5¢	50-100¢	25-50¢	--	1000	1000
Dynamic	2048	100-200	2.5/1¢	25-50¢	10-20¢	--	300	600
Array (TI-TMS2000) 17 inputs, 60 product terms, 18 outputs, 8 flip-flops	5040	500-1000		5-10¢	1.5-3¢	1-2	1100‡	5000
Logic Circuits								
Small Scale Integrated Circuits (14-16 pin DIP) TTL		3		35¢	25¢	3**	10-20	
Medium Scale Integration (MSI) TTL		20-50		7-16¢	3-6¢	.5-1	5-20	
Custom LSI								
TTL		150		20¢	8¢	200	5-20	
MOS		500		4¢	2¢	60	20-100	

* Without develo*pment* cost.

† Dollars per gate equivalent for customizing

‡ Without clock

** PC Board layout or wirewrap program.

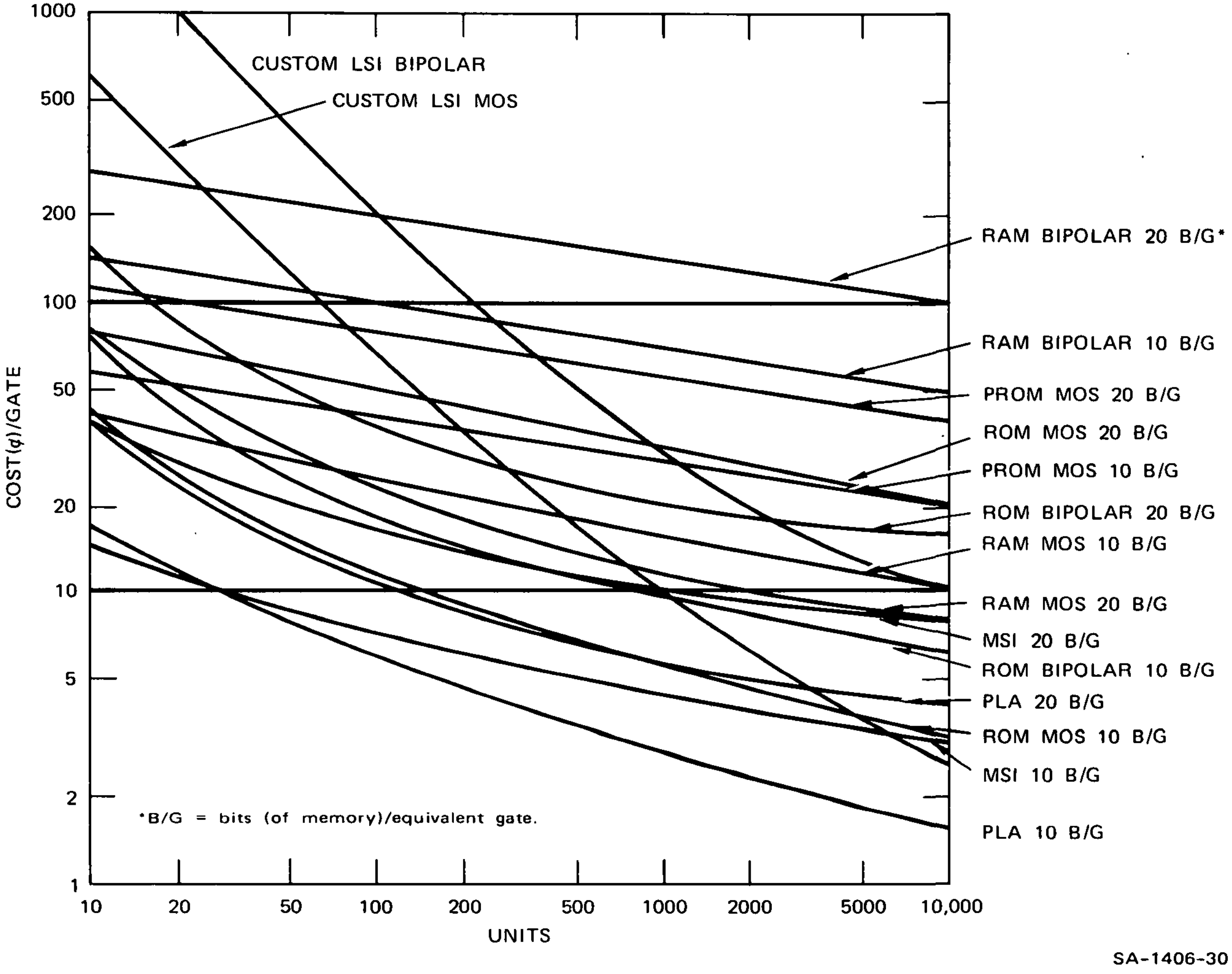


FIGURE 1 LSI COSTS FOR DIFFERENT TECHNOLOGIES

the 100 unit price should be used. On the other hand, if SSI is used for the design, a total of 150,000 SSI ICs may be used, mounted on 15 different types of PC boards. A unit number of 10,000 would probably be the proper place to check the gate cost for SSI in this case. In the example above, we would have found SSI at 25¢/gate to be preferred before the MOS LSI at 64¢/gate. If, on the other hand, more than approximately 500 systems were produced, the LSI version would become the cheaper one. MSI, in those places where usable circuits are available, successfully competes with MOS LSI up to unit volumes in the order of 10,000. A possible contender for the arbitrary logic is the Programmable Logic Array (PLA) which, at volumes below about 3,000 units, gives lower cost than alternative methods.

A direct comparison can be made between MOS LSI and MOS ROMs, as each system will have only one of each type, whether LSI or ROMs are used. Considering the case where 20 ROM bits are equivalent to one LSI gate, we see that if less than 140 systems are planned ROMs give lower cost per gate equivalent than the customized LSI version but that for a larger number of systems, the cost rapidly changes in favor of the LSI approach.

Circuit Performance

We will now turn our attention to the questions of speed, power dissipation, and cost. In view of the rapid introduction and acceptance of new process technologies, those that appear most promising are included in the following survey.

Table 2 illustrates gate delays, power and area requirements for gates with 3 inputs, and a fanout of 3 for a variety of circuit technologies and techniques. The reader is cautioned not to use the displayed data too rigorously, as in many cases the data has been derived by rather gross interpolations. The shown gate area does not take into account the area for interconnection between gates separated by one or more gates. The area for interconnect has, however, been assumed to be twice the gate area in calculating chip area. In other words, chip area = 3 X n X (gate area), where n is the number of gates. The area required for pads, protective devices, and output drivers is also assumed to be accounted for in the total.

Table 2

Performance of Gate with 3 Inputs and Fanout of 3

	Ratioed Logic							Dynamic 2φ Logic, Separate Power					Dynamic 2 & 4φ with Power Clock				
	Gate Delay	Static Power Dissipation	Gate Area	Chip Size	Gates/Chip Mixed Logic	Power/Chip (Static)	Power/Gate Dynamic µW/MHz	Gate Delay*	Dynamic Power µW/MHz/Gate	Gate Area	Chip Size	Gates/Chip	Gate Delay*	Dynamic Power/Gate	Gate Area	Chip Size	Gates/Chip
Technology	ns	µW/gate	mil^2	$K\ mil^2$	gates/chip	mW/chip	µW/MHz	ns	µW/MHz	mil^2/gate	$K.mil^2$		ns	µW/MHz	mil^2	$K\ mil^2$	
Bipolar																	
Standard Buried Collector (low power)	40	2000	50	22.5	150	300	--										
Collector Diffusion Isolation	6	4000	40	22.5	180	720	--										
Base Diffusion Isolation	40	1000	50	22.5	150	150	--										
"Trimask"	50	200	10	22.5	750	150	--										
Isoplanar	10	2000	20	22.5	370	740	--										
Schottky Clamped TTL	3	15000	50	11	66	1000	--										
MOS																	
High Voltage P-Channel	200	2000	12	18	500	1000 mW	50	80	100	40	40	330	60	60	30	40	445
Low Voltage P-Channel	140	1000	8	24	1000	1000 mW	40	60	70	30	40	445	40	50	25	40	530
Silicon Gate P-Channel	100	1000	6	18	1000	1000 mW	40	50	70	25	40	530	35	50	20	40	670
Ion Implanted P-Channel with Depletion Devices	80	800	8	30	1250	1000 mW	30	--	--	--	--	--	--	--	--	--	--
N-Channel Devices Metal Gate	100	1000	8	24	1000	1000 mW	40	40	70	30	40	445	30	50	25	40	530
Silicon Gate N-Channel	80	800	6	22.5	1250	1000 mW	30	30	70	25	40	530	25	50	20	40	670
Ion Implanted N-Channel with Depletion Devices	60	700	8	34	1430	1000 mW	30	--	--	--	--	--	--	--	--	--	--
Orthoplanar N-Channel with Depletion Devices (5V)	50	300	4	40	3300	1000	20	--	--	--	--	--	--	--	--	--	--
Complementary MOS	100	< 1	20	40	700	< .7	40	--	--	--	--	--	--	--	--	--	--

Usually limited by clock driver and clock skew.

The chip area is limited either by the largest chip area that can be produced with today's photolithographic techniques and with acceptable yields or by the maximum acceptable power dissipation.

The maximum chip size in the table was chosen to 200 X 200 mil^2= 40,000 mil^2 for MOS and 150 X 150 = 22,500 mil^2 for bipolar. The maximum power per chip was set to 1 W, which is a rule of thumb used in the industry for a 40 lead ceramic dual in-line pack. The gate delays are average delays expected in a mixed type of logic, with the capacitance of the interconnection on the chip included. In reviewing the column of gate delays we find, as expected, that the bipolar technologies give the shortest delays. The Schottky diode clamped TTL is superior with its 3 ns delay, but the penalty in power dissipation is still high, limiting the number of gates per chip to 60. Much lower power can be achieved, but not without sacrificing speed. The standard low power TTL technology gives gate delays in the order of 40 ns with a limitation to 150 gates/chip, on account of the 50 mil gate area. As a comparison, ratioed MOS gates have delays varying between 50 and 200 ns with area requirements of 4 to 12 mil^2. The smaller gate area required and the larger chip sizes give gate counts from 500 to 3300 per chip. The orthoplanar n-channel technology with depletion mode load devices is most interesting, in that with 50 ns gate delay it can operate on a +5 volt supply with low power consumption, 300 µW, and small geometry, 4 mil^2/gate, allowing up to 3300 gates/chip. The complementary MOS is superior in its low stand-by power consumption. The dynamic power consumption at the operating frequency is, however, of more relevance in the logic of an on-line computer. This power is consumed in charging and discharging capacitive loads in the circuitry. The dynamic dissipation is a linear function of the operating frequency. The dissipation at 1 MHz is given in the table and is of the same order of magnitude for both regular MOS and CMOS.

The dynamic logic, 2 phase, 4 phase, with or without power clock, for a given MOS technology gives potentially shorter gate delays than the ratioed logic, but the area per gate tends to increase. Gate delays as low as 25 ns can be achieved, provided the clock driver can meet the requirement. Practical circuits so far have, however, been limited by clock rates in the order of 1-5 MHz. This is somewhat misleading, as several levels of logic are typically performed during a clock cycle. Depletion devices, as loads and complementary logic, will reduce the importance of the dynamic circuit techniques. A more detailed comparison of these techniques is recommended. More information on both the orthoplanar and a CMOS silicon gate technique is expected to help in this evaluation.

The Programmed Logic Array (PLA) may be useful for the proposed NASA applications for small production quantities because of the small number of duplicate parts needed. Although PLA units can be made in various technologies, the CMOS technology is believed most desirable, since it combines good speed/power trade-off with excellent noise immunity. The speed of PLA units is somewhat lower than the speed of equivalent gate networks.

Although PLA is applicable to many of the arbitrary logic requirements in the proposed class of systems, specialized custom LSI circuits will also be needed for some areas of the arbitrary logic, particularly in the area immediately associated with the central processor units. CMOS is also the preferred technology for the custom LSI circuits

III MEMORY HIERARCHY

In considering the memory requirements, we distinguish a hierarchy of memories to carry out different functions. The principal function of possible technologies are shown in Table 3. Selection of the appropriate technology will be based on reliability, cost and speed.

Reliability

The reliability of a fault tolerant system is directly related to the reliability of its components. For given system requirements, therefore, the more reliable component can permit system simplifications not possible for less reliable components. Core memories are the traditional working memories for computers. These memories have a relatively high reliability within the limited range of permitted operating temperatures. The reliability of a core system is affected mostly by the number of solder connections in the array wiring and by the number of active and passive components used for driving and sensing. The reliability of the core arrays themselves is rather high after initial elimination of bad cores. The drivers are designed to operate at high current and high voltage and are, therefore, exposed to higher than normal electrical stress. Some degree of integration has reduced the component count in the core memories, but a rather large number of components still remain. We estimate the mean time to failure for the Ampex 1885 memory (8K × 18 bits) to be 8,000 hours (excluding power supply). This memory uses 107 ICs, 172 transistors, 236 diodes and 48 diode packs with 16 diodes in each, a total of 563 semiconductor components with from 2 to 16 lead packages.

This should be compared to an MOS memory based on chips, each holding 4096 bits of memory with decoding and sensing all included on a chip. This memory would require 36 MOS chips + 4 clock drivers +, at most, 25 more ICs for buffering and chip decoding (assuming TTL for this purpose), a total of about 65 ICs would be sufficient. Also, the electric stress on the MOS memory components, with the possible exception of the clock drivers, is considerably less than on the IC components of the core memory.

Table 3

Memory Hierarchies

Unit	Functions	Candidate Technologies
Micro-program Control store	Micro program steps, Initial bootstrap loader	Semiconductor RAMS, ROMS, PROMS
High Speed Cache stores	Data,Instructions	Semiconductor
Main Processing Store	Data ,Instructions , Input/Output Buffers Buffers	Core,Semiconductor, Plated wire
Back-up (Read only) Store	High Criticality Program Store (e.g. Flutter Control)	Semiconductor ROMS, PROMS, Plated wire
Read/Write Back up Store	In-flight Data Recording, Back-up airspace data (runway configurations etc)	Tape, Cassette, Disk, Drums
Other		Bubble Memories, Charge-coupled Devices, Soniscan, Domain Tip

The standard operating range for core memories is $0^{\circ}C-50^{\circ}C$, while the MOS memories can operate between $-55^{\circ}C$ and $+85^{\circ}C$. The failure rate on MOS arrays is estimated by one manufacturer (AMI) to be between 0.01% per 1000 hours and 0.1% per 1000 hours, depending on complexity and size. This corresponds approximately to figures quoted for bipolar ICs. Using the higher failure rate figure for the 65 chips would give a failure rate of 6.5% per 1000 hours, or a mean time to failure of 15,400 hours. The figure can be improved by using MOS LSI also for the control and buffer circuits.

The most common argument in favor of core memories versus MOS memories is that the core is a basically non-volatile memory element, while the MOS memory loses its content when power is removed. The stand-by power consumption of an MOS memory is as low as 1 W per million bits for P-channel silicon gate dynamic memories and lower than that for the 4096 bits/chip n-channel silicon gates just being announced. A 4096 X 16 bit memory could maintain its data with a battery supplying less than 100 mW. A 0.1 amp-hour, 10 volt battery would maintain the memory for 10 hours and would also reduce the effect of intermittent power failures, if used both for memory and logic. Such battery protected systems are presently appearing on the market.

A remaining concern is that the dynamic MOS memory may, if storing data for a long time, change its content due to noise pickup. This is certainly a possibility that applies to both types of memories when power is on and if insufficient care has been taken in the design to suppress outside noise. The core memory with power turned off is, of course, safe while the MOS memory in sustaining mode is still exposed. In the airborne application, the power off situation would be an exception adding only a fraction to the noise susceptibility of the MOS memory.

Suppression of outside noise in the small volume MOS memory with backup power source is, however, simpler than for the support circuits in the core memory.

In summary, we therefore conclude that an MOS memory can have a longer mean time to failure by at least a factor of 2 and can, with backup battery pack and careful design, be no more volatile than a core memory.

Cost

Failure resistant systems will have some form of redundancy also in the memory area. Low cost of the memory is, therefore, as important here as elsewhere.

OEM manufacturers negotiate today with the semiconductor houses at cost per bit in the range 0.25-0.5¢ for MOS IC chips. It is safe to expect bit costs on the system level of less than 1¢ by 1974 and less than 0.5¢ per bit by 1975. This compares to OEM prices of 2¢/bit in MOS systems today and 1.5¢/bit in core memory systems of small to medium size. Initial pricing will be based on the advantages compared to core memories. After enough competition between MOS memory manufacturers is established (1974-75), memory prices will reflect the actual manufacturer's cost.

Speed

Memory cycle times in the order of 1 μs typical for minicomputer core memories would, in all probability, be sufficient for this application, especially as a system with several processors operating in parallel would relieve some of the speed requirements. Only in a worst case situation, when all spare processors are out of operation concurrent with peak calculations, would faster operation be required. (Our reliability estimates show that this event occurs with extremely low probability.)

There is a tradeoff between cost and speed in MOS memories in that device area tends to increase for higher speed and, thereby, reduce the number of bits per chip. The silicon gate n channel memories with 4096 bits/chip will have cycle times in the order of 500 ns and less with access times of less than 300 ns. This should be representative for what can be expected without pushing speed at the expense of area.

Memory Hierarchy for Airborne Computer

The memory hierarchy will naturally depend on the final organization of the system, but we can now illuminate some alternatives.

Starting with the processing units themselves and referring to Table 3, we will probably require a read-only memory for the micro-control of instructions. The built-in ROM will also have a program sequence to be used for loading the initial programs to the working memory from one of several possible input devices. This program loading would normally take place on ground, but could also be performed in flight via the communication interface. When programs are stored in redundant memories and if loss of data in one of the memories is suspected, transfer from other units may be another method of loading.

There may be a few programs of extremely critical nature, such as for flutter control and possibly for executive control that require an extra margin of safety. These programs may be permanently stored in one or more read-only memories.

The equipment for accumulation of in-flight data for later use in maintenance and failure analysis is also required. The means of storage is highly dependent on the volume of data to be stored. Tape recorder, cassette or loop, drums or disks are candidates for recording high volumes of data at low cost/bit. All have moving parts and are sensitive to mechanical shocks. Ruggedized designs have been used on-board aircrafts before and in some more severe environments than the planned application. One example is a disk memory from Librascope that has been used in airborne applications. The cost per bit of that memory at a capacity of 350,000 bits is 0.57¢. This is approximately the same price that MOS memories will be available by 1974. At 700,000 bits of memory, the same disk would reduce the bit price to 0.33¢. It would probably be 1975 or 1976 before high capacity MOS memories could be bought at that price. The sharing of power supplies, control circuits and clock drivers tends to improve the economy for large MOS memories.

The disk memories have, of course, a distinct advantage in the sharing of a common mechanism as long as capacity can be increased by adding heads only. The optimum case is always when all head positions are filled and used. The optimum case of the disk mentioned above would be a capacity of 7M bits and 100 heads at a cost per bit of 0.09¢. Figure 2 compares the cost per bit as a function of memory capacity for the disk memory discussed above with that of MOS memories 1972, 1973 and 1975. The rate of decrease per bit of the MOS memories will, by 1975, still be high compared to the change of cost for the disk memories. By 1975 it will be more economic to use MOS memories than disks up to 600-700K bits of memory. The crossover

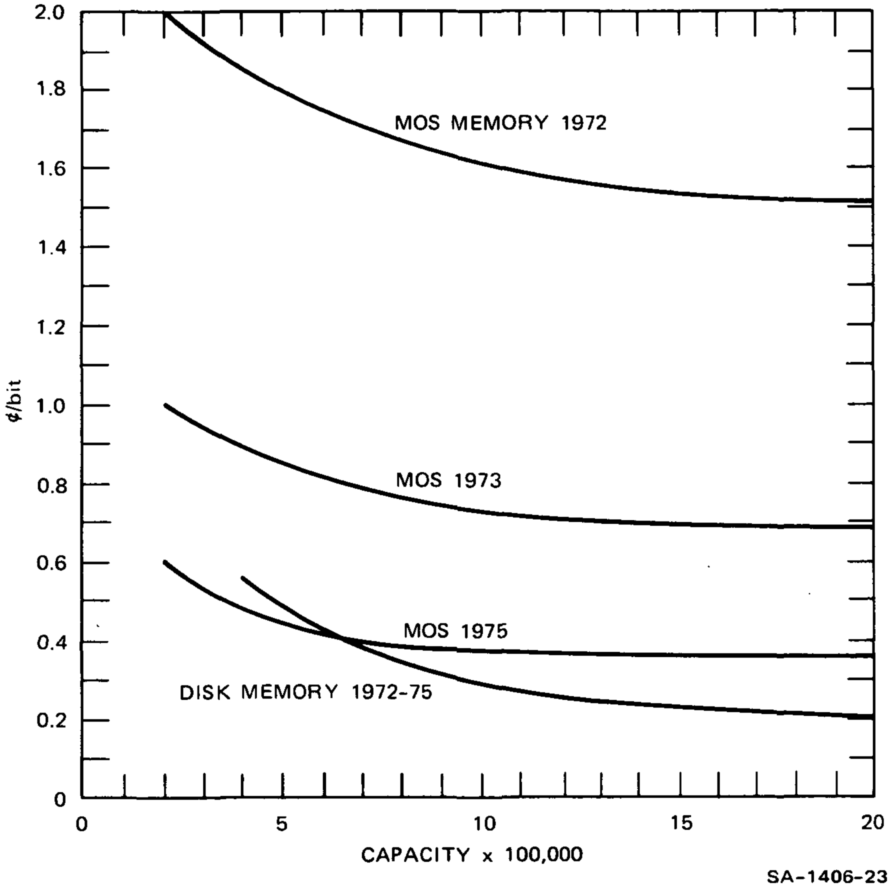


FIGURE 2 BULK MEMORY COST COMPARISON — DISK AND MOS

point will, in later production years, move even further to the right. The advantages of using MOS memories are low weight, small dimensions, and convenience for expansion.

One memory technology that has attracted attention and has been used in aerospace applications is the plated wire memory technique. Stated advantages are: non volatility, lower power and voltage requirements, and shorter access times than for core memories. Production techniques have been improved, making wires down to 2-3 mil diameter feasible. In spite of remarkable cost reductions, the costs are still close to ten times higher than those for core and MOS memories, a disadvantage that will be even worse relative to MOS memories in the next few years. The higher cost of this memory may, however, be acceptable to a user with strong enough belief in this specific technique. However, we recommend the more economical and, with reference to our earlier discussion of volatility, equally or more reliable MOS memory.

Other advanced memory techniques such as bubble memories, charge-coupled devices, Soniscan, domain tip, etc., are all very interesting candidates for future systems, but will, in our opinion, not be ready for incorporation in the earlier systems based on this project. A flexible system organization must, however, allow the inclusion of new technology, when it is practical and competitive.

Memories for the proposed class of machines should be designed using semiconductors, rather than magnetic cores. The semiconductor memories will, almost certainly, use MOS devices and will consist primarily of random access structures with some use of shift registers for I/O and inter-processor buffering. Operating power for the memories will be decreased below present levels, so that less than 1 μW per bit will be required while the memory is "idling" and less than about 50 μW per bit while the memory is active at the normal bit rate. High-speed memories will be constructed on insulating substrates such as sapphire within 5 to 10 years and the use of insulated substrate memories is recommended.

IV INTERCONNECTION TECHNOLOGY

As work on computer architecture progresses, it becomes clear that to achieve the reliability goals, several independent processors would be reconfigured rapidly on the occasion of a failure, so that the computational task at hand can proceed without error or undue delay. From the standpoint of physical configuration, it is desirable to package each processor on a separate card* for ease of manufacture, test and maintenance. Having each processor a separate entity also makes it easier to guarantee that the processor function is indeed independent. In addition to processor cards, there will be other cards for other functions, such as input/output circuits and power supply.

Semiconductor chips within each card structure may be packaged in subgroups for memory or other functions.

The group of cards comprising the computer system will probably be located in the same enclosure but elements of the input/output circuitry may be located in other enclosures mounted some distance away in the aircraft.

The following discussion emphasizes the requirements for data paths but it should be realized that appropriate interconnection networks are required also for power supply leads and control signals.

Data Path Requirements

In the following discussion, we consider the number of paths required for data and program transfer, the speed required for the paths and various means available for providing data paths that are reliable and economical. Three different levels of data path interfaces are considered:

* The term "card" is used to describe each major subassembly.

The "A" level, consisting of connections from chip to chip within a card; the "B" level, comprising the card-to-card connections, and the "C" level, containing connections from box to box and connections to points outside the main computer system.

Chip-to-Chip Connections, "A" Level

A processor assembly will consist of LSI chips for memory and for the arithmetic unit, hybrid ICs for clock generators and power supply regulators and some discrete devices. Even though the processor is complex, chip complexity in a 1975 design will also be high, so that relatively few chips will be needed for the memory, processor and the input/output circuits. We assume 2,000 gate complexity for a processor and 32K bytes (256,000 bits) for the memory.

If a complexity for the memory of 4,096 bits per chip is assumed, then 64 memory chips would be required. The number of pins required for data and other interconnections is shown in Table 4 for the cases of single-bit data (4,096 words) and 8-bit data (512 words).

Since the non-memory circuits in a semiconductor memory system contribute approximately 30 percent of the complexity, the total pin count can be obtained by increasing the memory numbers by 30 percent. Thus, the single bit organization would require 1.3 X 1280 or about 1,700 pins and the 8 bit organization would require about 1.3 X 1984 or approximately 2,600 pins. To compare this with a core memory, we must consider the total number of wire connections (soldered, wire wrapped, etc.) in such a memory. Consider the core memory to be composed of 64 core planes each containing 4,096 bits arranged as a 64 X 64 array. Representative figures for required connections are shown in Table 5.* Even allowing for reduction of connections by using alternative construction techniques, the total number of connections

* The use of special fabrication techniques such as "folding" can reduce the number of connections required, but not by a sufficient amount to invalidate the conclusions reached.

Table 4 - Connectors in LSI Memory

Function	Single-bit data	Multiple (8) bit data
Data	2 X 1 = 2	2 X 8 = 16
Address	12	9
Clock, Power & Control	6	6
Pins/Chip	20	31
Total Pins for Memory	1280	1984

Table 5 - Connectors in Core Memory

		Number of connections
Per Plane	X connections (both edges) Y " Sense line	128 128 2
	Total	258
Per Memory of 64 Planes	Item above (X 64) Decode, control, clock, etc. (estimated)	16K 1K
	Total	17K

is an order of magnitude higher than for the LSI case shown in Table 4. Memory failure due to faults in the connections will therefore be significantly less for LSI than core memories, assuming the same reliability for the connections.

Card-to-Card Connections, "B" Level

If data is transmitted in parallel groups from card to card, each card might typically have eight data input leads and eight data output leads. In addition, approximately six leads would be required for power supply and control and another ten leads for test purposes. A total of 32 leads would suffice for a parallel organized system. If serial organization is used, only one lead in and one lead out would be required for data paths and four leads for test. Adding six leads for power supply and control brings one to a total of 12 connections required. This assumes asynchronous (clock-free) serial operation.

If ten parallel-connected cards are used in the computer box, then about 320 total connections would be needed from card to card and about 160 would be required for data paths. If serial organization were used, only about 120 total connections need be provided and only about 20 data paths.

These estimates assume a non-redundant data path system. Redundancy will probably be required so that the number of data paths may double or triple. This factor tends to favor serial organization for inter-card data.

Box-to-Box Connections, "C" Level

The computer input/output channels represent a mixed variety of signals since these channels communicate with a variety of sensors and other "black boxes." These channels will probably be serial and perhaps 100 channels will be needed. If multiplexing is used on channels, the number of input/output data paths will be reduced since the same channel can be time-shared.

If auxiliary drum is used for storage of navigation data, there will be a special input/output channel required.

System Speed Requirements

The results of the other tasks suggest that the data paths must support communication between elements at speeds ranging from about 2 to 20 megabits per second. Since the computer system will be byte organized, a parallel communication structure would be required to operate at only about 1/8 of this rate.

Three general types of failures can occur in an interconnection network. They are faults of a permanent nature (such as a stuck flip-flop), temporary faults (such as those caused by noise or intermittent connections) and faults which can be either permanent or temporary but which cause permanent damage to other assemblies (a propagated damage fault).

The effects of these kinds of failures in the coupling element of an interconnection system depend on where the coupler is located in the interconnection hierarchy. At the primary level ("A"), the important effect is simple failure, since noise is not usually a factor. Intermittent failures, such as a poor bond, may result in an unfortunate but unavoidable decision to permanently disqualify an entire processor.

At intermediate levels of connection from card to card ("B" level), noise considerations become more important, since longer leads are involved, but propagated damage becomes very important. Propagated damage at the "B" level would, for example, result in permanent damage to one processor as the result of a failure in an adjacent processor.

At the highest level of interconnection from box to box ("C"), propagated damage could cause direct malfunction of devices in the input/output area. A temporary fault would cause another try to be made at a problem solution, for example, but would not seriously impair the long-term system performance.

Means of Providing Data Paths

We have considered eight different ways of communicating between various elements of the computer system. These range from simple wire connections at logic signal level to communications using fiber optics and optical couplers. None of the coupling means are applicable at all levels of the interconnection system hierarchy for providing data paths for reasons of economy and performance. The following comments are restricted to the eight means of providing data paths.

Wire-Logic Signal Level

Use of direct wires between elements on a card is the least expensive way to provide paths and can be done at logic signal levels, typically 5 volts. The speed can be as high as about 20 MB (mega bits per second) without requiring expensive interconnect structures. Providing a path between adjacent packages or chips would cost about 20¢ per path.

Wire-HNI Interface Circuits

When data paths extend from one card to another or from one assembly to another there is often difficulty with noise being coupled into the circuits through the wiring and causing bit errors. To alleviate these kinds of problems, HNI (High Noise Immunity) interface circuits such as line drivers and line receivers can be used between system subassemblies. Typically, this kind of circuit operates on a relatively high voltage supply of 10 to 20 volts and provides at least 5 volts of immunity to either outside interference or cross coupling between circuits. Circuits of this kind will cost about $1.00 per circuit or $2.00 for the path. Bit rate is usually restricted to about 10 MB (mega bits per second). HNI circuits also tend to reduce the susceptibility of systems to propagated damage.

Wire/LED/PT Devices

Isolators are now available using LED (Light Emitting Diode) light sources and PT (Photo-Transistor) units for transferring digital information through an optical medium. These devices are relatively inexpensive, about $2.00, and will operate at bit rates up to about 1/2 MB. The coupling has very high common mode noise rejection and excellent protection against propagated damage. For the system under consideration, the main restriction is that the data rate is low so that parallel connections must be used.

Wire/LED/PD

A LED source can also be used with photodiodes to allow bit rates up to about 10 MB and retain the other advantages of the LED/PT isolator.

Because the coupling efficiency is poor, amplifiers must be provided so that the cost is increased. Devices of this kind with amplifiers to restore levels to normal logic signals typically operate up to 10 MB and cost about $10.00.

Wire/Magnetic Coupling

Some systems have been designed using data paths consisting of wires to carry current and a magnetic coupling using a square loop core into the various system elements. This kind of coupling means is relatively inexpensive but usually limited to about 1 MB. Cost depends on how the assembly is connected but is about $3.00. Like the PT isolators, low speed requires parallel use in many system data paths.

A disadvantage of this kind of isolator is that dc logic levels cannot be transmitted, so that the data needs to be converted to a pulse form.

Fiber Optics/LED/PT or PD

It is possible to use a fiber optic bundle with a LED source in one box coupled by fiber optics to a data sink in another box. The light sensor can be either a phototransistor or a photodiode with corresponding changes and performance. With a PT sensor, data rates of about 1/2 MB are possible at a cost of about $5.00 for the sensors and LED source. With photodiodes the bit rate can be 10 MB but the cost is considerably higher due to losses in the optics and the lower sensitivity of the photodiode. Providing a data path in this manner would probably cost about $20.00 per path, not including the cost of the optic fiber.

Wire/Monolithic LED/PT Isolator

A recent development has allowed the construction of optically coupled isolators using monolithic construction so that both the LED source and the PT or PD sensor can be constructed on the same chip. Details of the device construction are still proprietary but performance is reported to be very good. Data rates of about 5 MB are attainable with costs predicted in the $5.00 to $10.00 range.

Table 6 lists five interconnection schemes believed to be most appropriate for this computer. The fiber optics isolators and the magnetic coupling method have been omitted from further considerations due to their relatively poor cost/performance. It is reasonably clear that for chip-to-chip paths direct wire or PC board connections at the logic level will be adequate and relatively inexpensive. The only cost involved is the PC board connection. This, of course, assumes no data path switching. For the "B" level connections from card to card, connections at logic level would cost only about $1.20 and result in a cost of about 6¢ per MB. HNI circuits would cost about $3.20 per path or 32¢ per MB and provide a fair protection against propagated damage. The optical isolators are more expensive but offer nearly perfect protection against damage from power supply failure or test voltages.

In the "C" category, direct logic level connections are not feasible due to noise problems. HNI is possible and workable, resulting in a cost of about $6.20 per path plus cost of installation. This results in about a 62¢ per MB cost. The optical isolators are much more attractive in the "C" category due to the excellent protection obtained from both noise and propagated damage.

As mentioned earlier, the cost of this alternative is based on about 85 paths per card required at the "A" interface for serial organization, a total of 850 for the assumed system. This corresponds to $170 at 20¢ a path for the connections at logic level. If parallel organization is assumed, 6700 paths would be required for all the cards, a total cost of $1,340.

At the "B" interface, 160 paths would be needed for parallel-organized systems, and 20 paths for serial organization. Direct connection would be possible, but not recommended, so that HNI would be indicated at $3.20 each. The cost for parallel paths in the computer box would be about $284 and only $64 for serial circuits.

The advantages of optical isolation could be obtained by doubling these costs.

Table 6 - Connection Costs

Cost/Path/MB (cents)

			A Chip-Chip		B Card-Card		C Box-Box	
Data Path via	Speed MB	cost $	Unit Cost	¢/MB	Unit Cost	¢/MB	Unit Cost	¢/MB
Wire - logic level	20	--	(PC Bd 20¢)	1	A + $1 per conn. pair $1.20	6	NF	
Wire - HNI circuits	10	2	.20 + $2 $2.20	22	A + $1 $3.20	32	B + 2 $6.20	62
Wire - LED/PT isolator	0.5	2	.20 + $2 $2.20	440	A + $1 $3.20	640	B + $2 $5.20	1040
Wire - LED PD isolator	10	8	.20 + $8 $8.20	82	A + $1 $9.20	92	B + $2 $11.20	112
Wire - monolithic isolator	5	5	.20 + $5 $5.20	104	A + $1 $6.20	124	B + $2 $8.20	164

⬭ = path cost

As mentioned earlier, some form of data-path redundancy will probably be needed so that a final system design would require more than the above numbers of devices at the "B" interfaces. The box-to-box paths at the "C" interfaces could be provided by HNI circuits, but this would not provide protection from propagated damage. We have assumed that roughly 100 paths would be required from box to box, but some of these paths will be obtained via multiplexing and others will be direct. Data rates are expected to be below 5 MB in all cases, and much lower in some. The monolithic isolator would, therefore, be applicable at a cost of about $8 per path, or a total cost of roughly $800, since serial transmission will probably be used.

The preceding discussion shed some light on the options available for providing data paths in the computer at various levels in the structure. We can also estimate roughly the cost of providing these data paths by various means. The failure rate in the ordinary sense of the added circuits required in the data paths, such as HNI circuitry and optical isolators, is probably not going to adversely affect the system reliability. The value of providing some feature such as freedom from propagated damage is more difficult to determine. Certainly one wants to be assured that each of the processor cards in the system is independent and will not be affected by other failures, either in adjacent processors or in other boxes in the aircraft. HNI circuits can do a reasonably good job at the "B" level but are believed inadequate at the "C" level. Propagated damage is much less likely with HNI circuits but is almost inconceivable with optically coupled circuits. A factor which strongly affects the cost of providing data paths is whether or not the system is organized serially or in parallel.

Data paths within the proposed computer will use different technologies depending on the hierarchy in the interconnection system. For chip-to-chip connections, we recommend conventional printed circuit boards with special precautions to minimize crosstalk and signal distortion due to the high bit rates required. For card-to-card connections, the optical coupler would be the recommended method for providing data paths, since propagated damage is prevented by their use. For box-to-box connections, optical couplers are also recommended and fiber optic devices may be substituted for optical couplers within five to ten years.

Reliability

The proposed computer system will have a very large number of semiconductor devices and most of the devices will be used for memory. Approximately 80% of the semiconductor circuits will be used directly for memory or associated with the memory function. The processes used for producing the circuits must be mature and well understood. In addition, extensive use should be made of testing the finished circuits and aerospace level inspection should be enabled where practical. At present, the failures observed in semiconductor systems are usually dominated by interconnect failures at the chip level. By careful quality control procedures, the failures due to chip level interconnects can probably be reduced to about 10^{-8} failures per hour. However, the internal failures in the device structure will also contribute significantly to the overall failure rate, so that the failure rate at the chip level will be about 10^{-7} per hour due to all causes.

For the first few hundred hours after the system is assembled, higher chip failure rates can be expected, but since the system is fault-tolerant, this should not be serious; only operating costs are affected.

Although the logic circuits represent only about 20 percent of the chips, more effort will be required for reliability assurance, due to the variety of circuits needed, compared to the more standardized chips used for memory. Power supply and input/output circuits will also require more effort than the memory, since the number of circuit elements is smaller, and in the power supply case, higher power devices are needed. These circuits should receive extensive "burn-in" prior to flight use.

V SUMMARY AND RECOMMENDATIONS

This portion of the project work was divided into three areas: logic circuits in the processor and subsystems, memory technology, and interconnection technology. As the work on computer architecture progressed, it became clear that a redundant system using a multiplicity of identical processor subsystems would be required; this departure from conventional computer architecture imposed a new set of constraints on logic, memory and interconnection technology.

The arbitrary logic in the processor and input/output subsystems can be done in a number of ways. The Programmed Logic Array (PLA) appears to be most effective for low qualities of computer systems, provided that their functional characteristics are compatible with the logical organization of the processor. For some of the arbitrary logic areas, however, specialized custom LSI circuits will also be needed. The preferred technology for the PLA chips and for the custom LSI chips is CMOS.

Most of the computer memories now in use employ magnetic core technology. However, the recommended architecture for the computer system uses distributed memory, which favors a semiconductor memory approach rather than the use of magnetic cores or magnetic wire. In addition, both Read-Only Memories (ROM) and Programmable Read-Only Memories (PROM) are available using semiconductor technology. The PROM and ROM structures allow storage of essential information for system start-up in the same manner now provided by magnetic memories. For data storage during power interruption or at brief out-of-service intervals, low power shift register memories are available which can be operated on small batteries for several hours, even though the system power is unavailable. The proposed all semiconductor system, therefore, will provide higher speed performance than magnetic memories and, in addition, allow smaller memory assemblies.

Both shift register and random access memories will be used (the latter dominating) and will be based on MOS device technology. For the high-speed random-access memories associated with the processor chips, an insulated substrate technology using materials such as sapphire with high-speed CMOS circuits is preferred.

Data paths within the computer will use different technologies depending on the hierarchy in the interconnection system. On each subsystem circuit card, the chip-to-chip connections will be best accomplished by using conventional multi-layer printed circuit boards with special precautions to minimize crosstalk and signal distortion at the high bit rates required. The data rate for card-to-card connections will be lower and timing will be less critical. However, noise is a more serious problem from card to card and there is also the possibility of provocated damage. For these reasons, the use of optical couplers is recommended for card-to-card data paths. For box-to-box connections from the computer to other portions of the system, optical couplers are also recommended and are available at adequate data speeds and with excellent solution characteristics. Towards the end of the period of interest, however, optic communications instead of wire or coaxial cable couplings will be available and their use should be considered.